

Building Virtual Machine Labs: A Hands-On Guide (Second Edition)

By: Tony Robinson

Building Virtual Machine Labs: A Hands-On Guide (Second Edition). Copyright © by Tony Robinson

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means electronic or mechanical, including photocopying, recording, or by any information or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13: 9798463249678

Cover Illustration: Stella Fin (<https://www.stellafin.com/>)

Editor, Technical Reviewer: Tony Robinson

Product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of this work, the author shall not have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

Contents

Foreword: Once More unto the breach.....	17
Chapter 1 Patch Notes	19
Chapter 1: If you build it.....	20
1.1 Who is this book for?	20
1.2 Getting the Most out of this Book	20
1.3 Notation	21
1.4 A Note About Software Versions, and The Three Rules of IT Disciplines	21
1.5 Software Recommendations.....	22
1.5.1 Windows Software Recommendations.....	24
1.5.2 MacOS Software Recommendations	24
1.5.3 Linux Software Recommendations	25
1.5.4 Operating System Installation Images	26
1.5.5 Register accounts on these websites.....	26
1.6 Linux users, MacOS users and the which command	27
1.7 Linux Users and Kernel Headers	28
1.7.1 How to Acquire Kernel Headers for Ubuntu/Debian-based Distributions	29
1.7.2 How to Acquire Kernel Headers for Redhat Enterprise/CentOS-based Distributions.....	30
1.8 Using Compression Tools	33
1.8.1 7-Zip on Windows	33
1.8.2 Finder on MacOS.....	34
1.8.3 zip/unzip and gzip/gunzip on Linux (and MacOS)	34
Chapter 2 Patch Notes	36
Chapter 2: Recommended Skills and Knowledge	37
2.1 TCP/IP Networking	37
2.2 Navigating Operating Systems, and their Installation Procedures	38
2.3 Recommended Training Resources	38
Chapter 3 Patch Notes	40
Chapter 3: Virtual Machines and Hypervisors	40
3.1 What is Virtualization?	40
3.2 What is a Hypervisor?	41
3.2.1 Hosted Hypervisors	41
3.2.2 Bare-metal Hypervisors.....	42

Chapter 4 Patch Notes	44
Chapter 4: Introduction to Virtual Networks: Hosted vs. Bare-metal Hypervisor Networking	44
4.1 Hosted Hypervisor Networking – Host-Only, Bridged, and NAT Network segments	44
4.1.1 Bridged Networking	45
4.1.2 NAT Networking (and Port Forwarding)	46
4.1.3 Host-Only Networking.....	48
4.1.3.4 Virtual Network Adapters	48
4.2 Bare-metal Hypervisors and Virtual Switches.....	50
Chapter 5 Patch Notes	51
Chapter 5: Hardware.....	51
5.1 RAM.....	51
5.2 Disk I/O.....	52
5.2.1 Hard Disk Drives	53
5.2.2 Solid-state drives.....	53
5.2.3 RAID arrays.....	53
5.3 CPU Cores and Features.....	54
5.4 Virtualization Extensions (AMD-V, Intel VT-x)	54
5.5 Performance as a Vicious Feedback Loop.....	56
Chapter 6 Patch Notes	57
Chapter 6: Virtual Lab Design and Overview	57
6.1 Lab Network Description – Virtual Machines	59
6.1.1 pfSense.....	59
6.1.2 SIEM	59
6.1.3 IPS.....	60
6.1.3.1 AFPACKET bridging, and Fail-Closed Networking	60
6.1.4 Kali.....	61
6.1.5 Metasploitable 2	62
6.2 Lab Network Description – Network Segments.....	63
6.2.1 Bridged (Physical) Network.....	63
6.2.2 Management Network.....	63
6.2.3 IPS 1 and IPS 2 Networks	63
6.3 Resource Allocations, and Hardware Requirements	64
Chapter 7 Patch Notes	67

Chapter 7: The Importance of a Password Manager	67
7.1 Benefits of Password Managers.....	67
7.2 Weaknesses of Password Managers.....	68
7.3 Mitigating the Weaknesses.....	68
7.4 Creating a Password Database File with KeePassXC.....	70
7.5 Creating Password Database Entries with KeePassXC.....	76
Chapter 8 Patch Notes:	79
Chapter 8: Time to Choose Your Destiny	80
8.1 Hypervisor Choices.....	80
8.2 Hypervisor Guide – Chapter Outline.....	82
Chapter 9 Patch Notes	84
Chapter 9: Client Hyper-V	86
9.1 Prerequisites	86
9.1.2 msinfo32.....	87
9.2 Installing Client Hyper-V	91
9.3 Customizing Client Hyper-V	92
9.3.1 Hyper-V Settings	92
9.3.2 Virtual Switch Manager.....	96
9.3.3 Configuring the Host-Only Network Interface (Management Virtual Switch).....	100
9.4 Building the First VM, pfSense	102
9.4.1 VM Creation	102
9.4.2 pfSense Virtual Machine Settings (Part 1)	109
9.4.3 First Boot and OS Installation.....	112
9.4.4 pfSense Virtual Machine Settings (Part 2)	117
9.4.5 pfSense Command-Line and initial interface configuration	120
9.4.5.1 The Assign Interfaces Wizard.....	120
9.4.5.2 Setting IP Addresses for WAN, LAN, and OPT1.....	124
9.4.6 Testing Internet Connectivity using Shell commands.....	131
9.4.7 Finish setting up pfSense	134
9.5 Create the Remaining Virtual Machines	135
9.5.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali.....	135
9.5.2 Operating System Installation.....	141
9.5.2.1 Installing Ubuntu on the SIEM VM.....	141

9.5.2.2 Additional Virtual Machine Settings – SIEM VM.....	149
9.5.2.3 Booting the SIEM VM for the first time	149
9.5.2.4 Installing Ubuntu on the IPS VM	154
9.5.2.5 Additional Virtual Machine Settings – IPS VM	158
9.5.2.6 Booting the IPS VM for the first time.....	159
9.5.2.7 Installing Kali Linux on the kali VM	161
9.5.2.8 Additional Virtual Machine Settings – kali VM	173
9.5.2.9 Booting the kali VM for the first time	173
9.5.3 Metasploitable 2	177
9.5.3.1 Converting the Metasploitable.vmdk to VHDX.....	179
9.5.3.2 Creating the Metasploitable 2 VM.....	182
9.5.3.3 Adjusting Metasploitable 2 VM settings.....	184
9.5.3.4 Booting Metasploitable 2.....	186
9.6 Checkpoints.....	189
9.6.1 How to Create a Checkpoint	189
9.6.2 Restoring a Checkpoint	192
9.6.3 Create checkpoints for the SIEM, IPS, Kali and Metasploitable 2 virtual machines	193
9.7 Chapter Review	194
Chapter 10 Patch Notes	195
Chapter 10: VirtualBox.....	196
10.1 Windows Installation Guide.....	197
10.2 MacOS Installation Guide	199
10.3 Linux Installation Guide	203
10.4 Customizing VirtualBox.....	208
10.5 Configuring the Host-Only Virtual Network Adapter.....	210
10.5.1 Setting the Host-Only Adapter's IP Address	212
10.5.1.1 Windows and <code>ncpa.cpl</code>	212
10.5.1.2 MacOS and <code>ifconfig</code>	214
10.5.1.3 Linux and <code>ip addr</code>	214
10.6 Building the first Virtual Machine, pfSense	216
10.6.1 VM Creation	216
10.6.2 pfSense Virtual Machine Settings (Part 1)	219
10.6.2.1 Virtual Machine Network Settings.....	224

10.6.3 First Boot and OS Installation	227
10.6.4 Virtual Machine Settings (Part 2).....	231
10.6.5.1 The Assign Interfaces Wizard.....	234
10.6.5.2 Setting IP Addresses for WAN, LAN, and OPT1.....	238
10.6.6 Testing Internet Connectivity using Shell commands.....	245
10.6.7 Finish setting up pfSense	248
10.7 Create the Remaining Virtual Machines	249
10.7.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali.....	249
10.7.2 Operating System Installation.....	255
10.7.2.1 Installing Ubuntu on the SIEM VM.....	255
10.7.2.2 Additional Virtual Machine Settings – SIEM VM.....	263
10.7.2.3 Booting the SIEM VM for the first time	265
10.7.2.4 Installing Ubuntu on the IPS VM.....	270
10.7.2.5 Additional Virtual Machine Settings – IPS VM	274
10.7.2.6 Booting the IPS VM for the first time.....	275
10.7.2.7 Installing Kali Linux on the kali VM	277
10.7.2.8 Additional Virtual Machine Settings – kali VM	285
10.7.2.9 Booting the kali VM for the first time	286
10.7.3 Metasploitable 2	289
10.7.3.1 Importing Metasploitable 2	291
10.7.3.2 Adjusting Metasploitable 2 VM settings.....	294
10.7.3.3 Booting Metasploitable 2.....	297
10.8 Snapshots.....	299
10.8.1 How to Take a VM Snapshot.....	299
10.8.2 Restoring a Snapshot	300
10.8.3 Snapshot the SIEM, IPS, Kali and Metasploitable 2 virtual machines.....	301
10.9 Chapter Review	302
Chapter 11 – Disclaimer for "M1" macs and macOS "Big Sur"	303
Chapter 11 Patch Notes	304
Chapter 11: VMware Fusion Pro.....	306
11.1 Installation	306
11.1.1 Permissions Dive	310
11.2 Virtual Network Editor	314

11.3 Configuring the vmnet2 Host Virtual Adapter	317
11.4 Building the first Virtual Machine, pfSense	318
11.4.1 VM Creation	318
11.4.2 Customizing the pfSense VM	322
11.4.3 First Boot and OS Installation	332
11.4.4 Virtual Machine Settings.....	336
11.4.5 pfSense Command-Line and initial interface configuration	337
11.4.5.1 The Assign Interfaces Wizard.....	337
11.4.5.2 Setting IP Addresses for WAN, LAN, and OPT1.....	341
11.4.6 Testing Internet Connectivity using Shell commands.....	348
11.4.7 Finish setting up pfSense	351
11.5 Create the Remaining Virtual Machines	352
11.5.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali.....	352
11.5.2 Creating Static DHCP Allocations for the SIEM, IPS and Kali VMs.....	357
11.5.3 Operating System Installation.....	358
11.5.3.1 Installing Ubuntu on the SIEM VM.....	358
11.5.3.2 Additional Virtual Machine Settings – SIEM VM.....	366
11.5.3.3 Booting the SIEM VM for the first time	366
11.5.3.4 Installing Ubuntu on the IPS VM.....	371
11.5.3.5 Additional Virtual Machine Settings – IPS VM	375
11.5.3.6 Booting the IPS VM for the first time.....	376
11.5.3.7 Installing Kali Linux on the kali VM	378
11.5.3.8 Additional Virtual Machine Settings – kali VM	386
11.5.3.9 Booting the kali VM for the first time	386
11.5.4 Metasploitable 2	390
11.5.4.1 Registering the Metasploitable 2 VM	390
11.5.4.2 Edit Metasploitable 2 Virtual Machine Settings	393
11.5.4.3 Metasploitable 2 Test Run	395
11.6 Snapshots.....	399
11.6.1 How to Create a Snapshot	399
11.6.2 Restoring a Snapshot	400
11.6.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines	402
11.7 Chapter Review	403

Chapter 12 Patch Notes	404
Chapter 12: VMware Workstation Pro	405
12.1 Installation	405
12.1.1 Windows Installation Guide	406
12.1.2 Linux Installation Guide	409
12.2 Customizing VMware Workstation	413
12.3 Virtual Network Editor	416
12.4 Configuring the VMnet1 Host Virtual Adapter	423
12.4.1 Configure the VMnet1 Host Virtual Adapter on Windows	423
12.4.2 Configuring the vmnet1 Host Virtual Adapter on Linux.....	425
12.5 Building the first Virtual Machine, pfSense	426
12.5.1 VM Creation	427
12.5.2 First Boot and OS Installation	437
12.5.3 Virtual Machine Settings.....	441
12.5.4 pfSense Command-Line and initial interface configuration	442
12.5.4.1 The Assign Interfaces Wizard.....	442
12.5.4.2 Setting IP Addresses for WAN, LAN, and OPT1.....	446
12.5.5 Testing Internet Connectivity using Shell commands.....	453
12.5.6 Finish setting up pfSense	456
12.6 Create the Remaining Virtual Machines	457
12.6.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali.....	457
12.6.2 Creating Static DHCP Allocations for the SIEM, IPS and Kali VMs.....	462
12.6.3 Operating System Installation.....	464
12.6.3.1 Installing Ubuntu on the SIEM VM.....	464
12.6.3.2 Additional Virtual Machine Settings – SIEM VM.....	472
12.6.3.3 Booting the SIEM VM for the first time	472
12.6.3.4 Installing Ubuntu on the IPS VM	477
12.6.3.5 Additional Virtual Machine Settings – IPS VM	481
12.6.3.6 Booting the IPS VM for the first time.....	482
12.6.3.7 Installing Kali Linux on the kali VM	484
12.6.3.8 Additional Virtual Machine Settings – kali VM	492
12.6.3.9 Booting the kali VM for the first time	492
12.6.4 Metasploitable 2	496

12.6.4.1 Registering the Metasploitable 2 VM	496
12.6.4.1 Upgrading the Metasploitable 2 VM	501
12.6.4.2 Edit Metasploitable 2 Virtual Machine Settings	503
12.6.4.3 Metasploitable 2 Test Run	505
12.7 Snapshots	507
12.7.1 How to Create a Snapshot	507
12.7.2 Restoring a Snapshot	509
12.7.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines	511
12.8 Chapter Review	512
Chapter 13 Patch Notes	513
Chapter 13: ESXi	515
13.1 Prerequisites	515
13.1.1 Installation Requirements	516
13.1.2 Hardware Compatibility	520
13.2 Installing ESXi	525
13.2.1 Acquiring the installation ISO	525
13.2.2 Downloading and Installing UNetbootin	528
13.2.2.1 Installing UNetbootin on Windows	529
13.2.2.2 Installing UNetbootin on MacOS	529
13.2.2.3 Installing UNetbootin on Linux	532
13.2.3 Using UNetbootin to create a bootable installer USB drive	536
13.3: Installing ESXi	539
13.4: Accessing the ESXi Web Interface	543
13.4.1: Configuring a Static DHCP Mapping for the ESXi Management Interface	543
13.4.2: Connecting to the ESXi Web Interface	548
13.5: Configuring ESXi	550
13.5.1 Assigning a License	552
13.5.2 Virtual Switches and Port Groups	553
13.5.3: Datastores	561
13.5.3.1: Staging	567
13.6 Building the first Virtual Machine, pfSense	569
13.6.1 VM Creation	569
13.6.2 First Boot and OS Installation	575

13.6.3 pfSense Virtual Machine Settings	579
13.6.3.1 Static IP Address/DHCP Reservation for the Bridged/WAN MAC Address	582
13.6.4 pfSense Command-Line and initial interface configuration	583
13.6.4.1 The Assign Interfaces Wizard.....	583
13.6.4.2 Setting IP Addresses for WAN, LAN, and OPT1.....	587
13.6.5 Testing Internet Connectivity using Shell commands.....	594
13.6.5.1 One Last Detail (enableallowallWAN).....	597
13.7 Create the Remaining Virtual Machines	599
13.7.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali.....	599
13.7.2 Operating System Installation.....	603
13.7.2.1 Installing Ubuntu on the SIEM VM.....	603
13.7.2.2 Additional Virtual Machine Settings – SIEM VM.....	611
13.7.2.3 Booting the SIEM VM for the first time	611
13.7.2.4 Installing Ubuntu on the IPS VM.....	616
13.7.2.5 Additional Virtual Machine Settings – IPS VM	620
13.7.2.6 Booting the IPS VM for the first time.....	621
13.7.2.7 Installing Kali Linux on the kali VM	623
13.7.2.8 Additional Virtual Machine Settings – kali VM	631
13.7.2.9 Booting the kali VM for the first time	631
13.7.3 Metasploitable 2	635
13.7.3.1 Acquiring the vCenter Converter Application.....	635
13.7.3.2 Converting and Uploading Metasploitable 2	638
13.7.3.3 Additional Adjustments	642
13.7.3.4 Uploading and Converting the Metasploitable VM without vCenter Converter Standalone	643
13.7.3.5 Final touches	647
13.7.3.6 Metasploitable 2 Test Run	655
13.8 Snapshots.....	658
13.8.1 How to Create a Snapshot	658
13.8.2 Restoring a Snapshot	659
13.8.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines	661
13.9 Chapter Review	663
Chapter 14 Patch Notes	664

Chapter 14: pfSense Firewall Policy and Network Services	664
14.1 The webConfigurator, and pfSense Setup Wizard	665
14.2 Checking for System Updates	671
14.3 Enabling Network Services.....	675
14.3.1 DNS Forwarding	675
14.3.2 NTP	682
14.3.3 Squid HTTP Proxy	685
14.3.4 DHCP	688
14.3.4.1 How to Create a Static DHCP Mapping	690
14.4 Firewall Policy	693
14.4.1 Firewall basics – Stateful Firewalls, Rule Order, and Implicit Deny Any	693
14.4.2 Firewall Aliases.....	696
14.4.3 Creating Firewall Rules.....	698
14.4.4 Firewall Rule Policy – Hosted Hypervisors	702
14.4.4.1 – WAN Interface.....	702
14.4.4.2 – LAN Interface.....	703
14.4.4.3 OPT1 Interface	705
14.4.4.4 Removing the Default Anti-Lockout Rule.....	708
14.4.5 Firewall Rule Policy – Bare-metal Hypervisors.....	711
14.4.5.1 WAN Interface.....	711
14.4.5.2 LAN Interface	713
14.4.5.3 OPT1 Interface	714
14.4.5.4 Removing the Default Anti-Lockout Rule.....	717
14.4.5.5 Removing the allow all pfSsh.php firewall rule.....	719
14.5 Chapter Review	724
Chapter 15 Patch Notes	725
Chapter 15: Routing and Remote Access for Hosted Hypervisors.....	727
15.1 Routing Tables and Static Routes.....	727
15.1.1 Persistent Static Routes on Windows	732
15.1.2 Static routes on Linux.....	733
15.1.3 Static Routes on MacOS.....	734
15.1.3.1 flightcheck-Linux and flightcheck-OSX.....	735
15.1.4 Enabling SSH access on Kali Linux	742

15.2 Remote Access for Windows Hypervisor Hosts	743
15.2.1 mRemoteNG	743
15.2.2 Creating Connection Profiles	745
15.2.3 Enabling Key-Based Authentication	750
15.2.3.1 Generating Public and Private SSH keys using PuTTYgen	750
15.2.3.2 Copying the SSH public key to lab VMs.....	760
15.2.3.3 Reconfiguring mRemoteNG to Use SSH keys.....	775
15.3 Remote Access for Linux/MacOS Hypervisor Hosts.....	784
15.3.1 The ssh command.....	784
15.3.2 Connection profiles and ~/.ssh/config.....	788
15.3.3 Enabling Key-Based Authentication	798
15.3.3.1 ssh-keygen.....	798
15.3.3.2 Copying the SSH public key to lab VMs.....	799
15.3.3.3 Testing Key-Based Authentication	808
15.4 Troubleshooting SSH Connectivity and Key-Based Authentication	809
15.5 (Optional Content) Remote Access Enhancements	813
15.5.1 Enabling SSH Access as the root User	813
15.5.1.1 Testing root SSH for Linux/MacOS Hypervisor Hosts	816
15.5.1.2 Testing root SSH for Windows Hypervisor Hosts.....	819
15.5.1.3 Remember, This isn't Strictly Necessary	821
15.5.2 Disabling password authentication over SSH	821
15.5.2.1 Backing Up (and Restoring) the /etc/ssh/sshd_config file.....	821
15.5.2.2 Modifying the PasswordAuthentication, ChallengeResponseAuthentication, and AuthenticationMethods directives.....	823
15.5.2.3 Verifying Password Authentication over SSH is disabled	826
15.6 Chapter Review	831
Chapter 16 Patch Notes	833
Chapter 16: Routing and Remote Access for Bare-metal Hypervisors	835
16.1 A Brief Review: Bare-metal Hypervisors vs. Hosted Hypervisors	835
16.1.1 Lab Network Design on Hosted Hypervisors.....	836
16.1.2 Lab Network Design on Bare-Metal Hypervisors.....	836
16.2 Introduction to Bastion Hosts	837
16.3 Creating A Bastion Host	839

16.3.1 Creating a Bastion Host Virtual Machine on VMware ESXi.....	840
16.3.2 Creating a Raspberry Pi Bastion Host.....	850
16.3.2.1 Prerequisites	850
16.3.2.2 Raspberry Pi Imager	853
16.3.2.2.1 RPI Imager Installation Instructions: Windows.....	853
16.3.2.2.2 RPI Imager Installation Instructions: MacOS	855
16.3.2.2.3 RPI Imager Installation Instructions: Ubuntu Desktop 20.04	856
16.3.2.3 Installing Raspbian using Raspberry Pi Imager	857
16.3.2.4 Booting the Raspberry Pi and Configuring Raspbian	861
16.3.3 Configuring Static Routes on the Bastion Host	867
16.3.3.1 Persistent Static Routes on Ubuntu, using netplan	867
16.3.3.2 Persistent Static Routes on Raspbian, using dhcpcd	873
16.3.4 Configuring the pfSense Firewall	878
16.4 SSH, SSH Tunnels, and You.....	882
16.4.1 SSH Tunneling Explained	882
16.4.1.1 Forward Tunnels, Illustrated.....	883
16.4.1.2 Reverse Tunnels, Illustrated.....	884
16.4.1.3 Dynamic Tunnels, Illustrated	886
16.4.2 Enabling the SSH service on the Kali Linux VM	888
16.5 Establishing SSH Connectivity to the Bastion Host and Lab VMs (Windows).....	889
16.5.1 Connecting to the Bastion Host with mRemoteNG	889
16.5.2 Enabling SSH Tunneling via PuTTY Session	891
16.5.3 Connecting to the SIEM, IPS and Kali VMs using Forward Tunnels	896
16.5.4 Generating SSH Keys for Key-Based Authentication (Optional)	898
16.5.5 Copying The authorized_keys File to the Bastion Host, and Lab VMs.....	903
16.5.5.1: Method 1 – WinSCP	903
16.5.5.2: Method 2 – Copy, Paste, echo, and file redirection	907
16.5.5.3: Method 3 – Copy and Paste, using vi	908
16.5.6: Creating and Modifying PuTTY Sessions to Enable Key-Based Authentication	910
16.5.7: Reconfiguring Connection Profiles, and Testing Key-Based Authentication	914
16.6 Establishing SSH Connectivity to the Bastion Host and Lab VMs (Linux/MacOS).....	919
16.6.1 The ssh command.....	919
16.6.2 Enabling and Testing SSH tunnels	920

16.6.3	Creating SSH connection profiles via <code>~/ .ssh/config</code>	925
16.6.4	Generating SSH Keys for Key-Based Authentication (Optional)	932
16.6.5	Copying The <code>authorized_keys</code> File to the Bastion Host, and Lab VMs.....	933
16.6.5.1	Method 1: <code>ssh-copy-id</code>	933
16.6.5.2	Method 2: <code>scp</code>	936
16.6.5.3	Method 3: Copy, Paste, and Output Redirection.....	938
16.6.6	Testing Key-Based Authentication	940
16.7	Troubleshooting SSH connectivity and Key-Based Authentication	942
16.8	Using the Bastion Host as a Web Proxy, using Dynamic Tunnels and FoxyProxy.....	946
16.8.1	Installation Instructions	946
16.8.2	Configuration Instructions	949
16.8.3	Adding a new proxy, enabling the proxy, and testing connectivity.....	951
16.9	(Optional Content) Remote Access Enhancements.....	959
16.9.1	Enabling SSH Access as the <code>root</code> User	959
16.9.1.1	Testing root SSH for Linux/macOS Users.....	960
16.9.1.2	Testing root SSH for Windows Hypervisor Hosts.....	966
16.9.1.3	Remember, This isn't Strictly Necessary	969
16.9.2	Disabling password authentication over SSH	969
16.9.2.1	Backing Up (and Restoring) the <code>/etc/ssh/sshd_config</code> file.....	969
16.9.2.2	Modifying the <code>PasswordAuthentication</code> , <code>ChallengeResponseAuthentication</code> , and <code>AuthenticationMethods</code> directives.....	970
16.9.2.3	Verifying Password Authentication over SSH is disabled	973
16.10	Chapter Review	977
Chapter 17	Patch Notes	979
Chapter 17:	Network Intrusion Detection	980
17.1	Making a Choice.....	981
17.2	Installing Snort3 (via Autosnort3).....	982
17.2.1	Confirming Autosnort3 success	986
17.3	Installing Suricata (via Autosuricata)	988
17.3.1	Confirming Autosuricata success.....	990
17.4	Troubleshooting Snort and Suricata problems	992
17.5	Chapter Review	994
Chapter 18	Patch Notes	995

Chapter 18: Setting up Splunk	996
18.1 Installing Splunk on the SIEM VM	996
18.1.1 Downloading Splunk Enterprise.....	996
18.1.2 Installing and Configuring Splunk (Part 1).....	1000
18.1.3 Installing and Configuring Splunk Enterprise (Part 2)	1002
18.1.3.1 Enabling SSL on Splunk Web	1002
18.1.3.2 Configuring a Receiver	1005
18.1.3.3 Switching to Splunk Free Licensing	1007
18.2 Installing and Configuring the Universal Forwarder on the IPS VM	1016
18.2.1 Downloading and Installing the Universal Forwarder package for the IPS VM	1016
18.2.2 Installing the Suricata TA	1020
18.2.3 Installing the Snort3 JSON Alerts App.....	1024
18.2.3.1 Installing Snort3 JSON Alerts on the SIEM VM.....	1024
18.2.3.2 Installing Snort 3 JSON Alerts on the IPS VM	1026
18.3 Restarting the Splunk Forwarder, and Testing Functionality	1029
18.4 Troubleshooting Recommendations.....	1032
18.5 Chapter Review	1035
Chapter 19 Patch Notes	1036
Chapter 19: End of the Beginning	1037
19.1 Chapter Review	1037
19.2 Remodeling and Expansion.....	1039
19.2.3 Outfitting a Malware Analysis Lab	1040
19.2.4 Outfitting an Offensive Security/Penetration testing lab	1045
19.2.5 Outfitting an Ops-Centric lab	1048
19.3 Final Words	1053
Chapter 20 Patch Notes	1054
Chapter 20: Extra Credit.....	1055
20.1 Hardening Hypervisor Security	1056
20.2 Update automation with the updater script.....	1074
20.3 Setting up ntpd on Linux lab VMs.....	1081

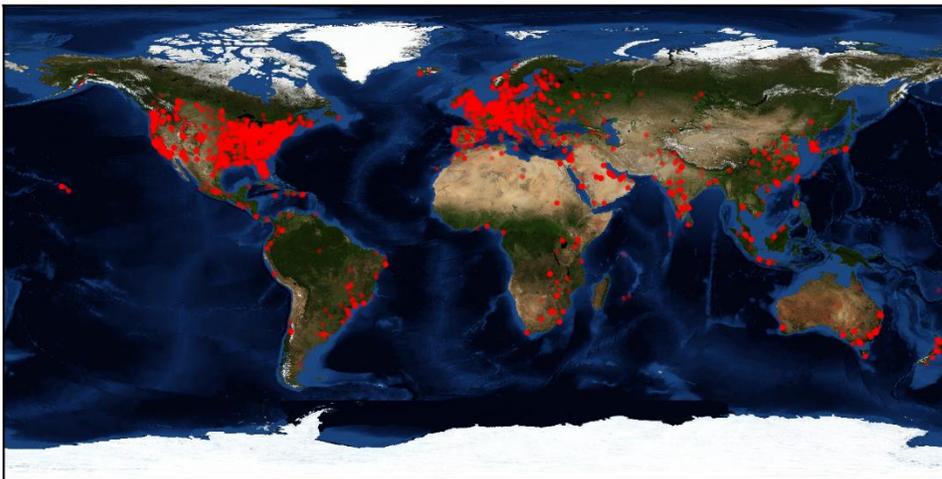
Foreword: Once More unto the breach...

So here we are again. Four years later, I'm embarking on a second edition of this book. When I originally wrote this thing, I had no plan and no intention of actually selling it as a textbook. I kind of just wanted to make sure that all of my virtualization knowledge was dumped somewhere because human memory is a terrible storage medium – at least mine is. Then I did something interesting and pulled some light statistical data, from the access logs on my web server:

```
for i in `grep [/url/to/my.pdf] access.log | cut -d" " -f1 | sort -u`; do whois -h whois.cymru.com $i | egrep -v ^AS >> results.txt; done
```

This set of commands greps the access logs for all requests to the URL where I hosted the PDF version of my book, cuts out the first field of the line, separated by spaces (the IP address), sorts them and removes duplicates, then feeds the output to the whois command. I tell whois to use the Team Cymru's WHOIS service¹, remove the header output for each result, and dump the collection of results to a text file. I then used a python tool to plot IP addresses by their geographical location on the world map.

It turns out that a book I thought of as nothing more as a brain dump of my virtualization knowledge was pretty popular. Tens of thousands of downloads from unique networks across the globe:



This image is an IP geolocation heatmap of the unique IP addresses that requested various copies of the in-progress version of this book when it was known as "Project Avatar".

It was downloaded by the intelligence community, Fortune 100 companies, higher education, and so many other interesting places. What was just an attempt to commit tribal knowledge gained over a career, became so much more to the IT community at large. After self-publishing, thousands of copies have flown off the shelf. I don't have a marketing team or budget, all I have

¹ <https://www.team-cymru.com/IP-ASN-mapping.html>

is a social media account, and an incredibly supportive community who have spread my work through word of mouth alone. To this day, people are still ordering both physical and digital copies.

So, what was the point of this long intro? It wasn't for me to brag. It wasn't for me to claim that I came from nothing and I did it all myself, because heavens know that is the farthest thing from the truth. I'm here to tell you that if some clown can commit his tribal knowledge gathered over the course of a career, transcribe it and dump it into a textbook, then successfully sell it around the globe, that there's a good chance you can do this too, and maybe even do a better job than I did.

I didn't get here alone. Not by a long shot, so let me offer some thanks where its due. First and foremost, I would like to offer my thanks to the technical reviewer and editor from the first edition, "Lord Gunter". This gentleman came to me out of nowhere, corrected my numerous grammar mistakes, and would often double check the technical content, inserting questions and comments of his own. Lord Gunter improved the first edition immeasurably. Without him, it would not be the body of work that it came to be.

Next, let me thank my employer, Hurricane Labs. What can I say other than thank you for the many opportunities to keep up with my writing over the last few years?

I'd also like to thank my wife and family for their support in my pursuit of this endeavor. Becka has endured so much endless grumbling whenever I discovered flaws in the work that I committed.

Finally, I would like to thank all of you out there for your patronage. Technology and IT books are pretty much a dime a dozen. Not to mention there are tons of more reputable writers and publishers out there. Even so, you gave my work a chance and together we've achieved something I never thought would happen: influencing a new generation of IT and security professionals. Good luck, have fun, and I hope you learn something interesting.

Chapter 1 Patch Notes

- Have you ever picked up a new edition of a book and wondered, “What's changed since the last edition?” Well, that’s what these “patch notes” are for.
- These patch notes will be included at the beginning of every chapter to describe significant changes from the first edition of the book.
- Added a small section to discuss the different types of font and notation the book uses
- Added a section with a list of software recommendations (for Windows, Linux and/or MacOS hypervisor hosts and/or management workstations), Operating system ISOs to download, and websites to register an account on to make downloading software and/or getting updates for portions of the lab easier.
- Added a section on how to install Linux kernel headers since most hosted hypervisors on Linux require them
- Introduced users to the joys and annoyances of the `sudo` command early
- Think of this as a section you can use to gather the required ingredients for the lab environment. That way, those who have slower internet service have a list of stuff to start downloading right from the start, while they're reading up on the introduction chapters.
- Added a section on using compression tools to decompress the pfSense and metasploitable 2 installation files.

Chapter 1: If you build it...

Have you ever found yourself with a new software tool, github project, automation suite, or a sample of malware you wanted to experiment with, while having some reasonable security measures in place to ensure that your experiments don't cause any problems? Well then, this is the book for you.

1.1 Who is this book for?

This book is designed to introduce readers to virtualization concepts, and provide instruction on how to build a secure and flexible lab environment with that knowledge. This lab environment that students build will provide a safe and secure network to practice information technology, cybersecurity, or other computer science concepts. Once readers have built the baseline lab environment, they will have the skill necessary to expand or reconfigure it in a way that better suits their needs.

1.2 Getting the Most out of this Book

Before we get much further, a word of advice: **this isn't a book that most people will want (or need) to read from front to back.** This book covers creating the same baseline lab environment across 5 different types of virtualization software (called *hypervisors*). This means that there will be a lot of repetition, with slight variations on the steps based on the hypervisor, the terminology associated with that software, and unique challenges that come with operating that particular hypervisor. Think of this book as one of those "Choose your own adventure" novels, where readers can pick and choose the content they want. Here are my recommendations for getting the most out of this book:

Read all of the first few chapters up to, and including Chapter 8, in order to gain a better understanding of the knowledge required to create the lab, understand how virtualization works, hardware and resource recommendations, and finally, to understand the how the baseline lab environment is designed before building it.

After chapter 8, There are then five chapters on how to perform the initial setup of your lab environment on one of five unique hypervisors:

- Oracle Virtualbox
- Microsoft Client Hyper-V
- VMware Workstation Pro
- VMware Fusion Pro
- VMware ESXi

These chapters, referred to as the hypervisor setup guides, will teach students how to acquire and install the hypervisor of their choosing, configure virtual networking, and other aspects

required to support the design of their virtual machine lab. These chapters will also provide guidance on the creation, and initial setup of the virtual machines themselves.

At the end of every hypervisor setup guide, there will be a section titled *Next Steps* that will point readers to other chapters to read in order to finish making the environment functional, as well as other chapters to consider reading in order to implement features that will contribute to a much better experience using the virtual machines.

1.3 Notation

Operating system commands and output will be rendered with the Consolas font.

Interesting bits of information, and important notes will be placed into boxes like this. They'll often be referred to as sidebar discussions, and will be more informal in tone, as I'm trying to address you directly about difficulties you may encounter, and how to deal with them as they come up.

Dialogue to pay attention to, such as the name of an application window, or the title of a configuration setting will be italicized.

Important Things to pay attention to will be underlined, in bold, and italicized

Links to other sections of this document (digital edition only) will be underlined

1.4 A Note About Software Versions, and The Three Rules of IT Disciplines

Writing books about IT disciplines is something of a daunting task due to how fast technology changes. The moment anyone attempts to put ink to paper about a given IT subject, something has inevitably changed, and the information contained in the book becomes deprecated almost immediately. The first edition of this book was produced in 2016, and in that time, all of the hypervisors have changed, as well as the operating system distributions and software used to build the lab environment in the first edition.

This book will refer to specific software versions for both the hypervisors as well as the operating system installation ISOs. These are the software versions that were available when this book was being revised. Don't obsess over having the exact same software versions. **Remember that it is a good security practice to keep software up to date.** This also includes hypervisor software, and virtual machines.

Of course, future versions of hypervisors and operating systems may differ in some way or another. Configuration settings may not be in the same place as an illustration says it should be, button colors and styles may have changed, radio buttons may now be checkboxes, and so on. This is because UI (User Interface) and UX (User Experience) developers changed where a particular setting or function is located, or how users are supposed to interact with it. Sometimes

this is done because they can't leave well enough alone, other times it's the result of feedback collected from the users to make their product better and easier to use.

If a configuration setting has moved, or it's not in a location indicated by the screen captures, **don't panic**. This is the first rule of most IT-related disciplines. When something doesn't go exactly according to plan, keep a level head. The second rule is that **software changes**. Sometimes, it seems as though that change is entirely arbitrary, while other times those changes are a legitimate improvement. The third rule is to **consult the documentation**. Very rarely are configuration options removed entirely due to a software update (however, it *does* happen sometimes, and is referred to as a feature regression). Maybe the configuration option has moved to a new menu location, or maybe it was integrated into another setting and no longer has an explicit configuration of its own. Consult the product patch notes, documentation included with the software (e.g., integrated help functions, vendor-provided documents online, etc.), as well as knowledgebases or forums about the software to see if there is anything noting where the configuration setting lives now.

This book isn't a guide to tell readers what buttons to click, and what menus to interact with, but to understand what the configuration settings accomplish, and why they are being modified. That way, students are able to further customize their lab environment on their own to better suit their needs.

1.5 Software Recommendations

In order to build and properly utilize the lab environment, there will be a lot of software students will be required to install. The purpose of this section is to provide a list of commands and software applications students will want to ensure are installed, and where they can be acquired if they don't already have them. This section will cover software recommendations for Windows, most Linux distributions, and MacOS. Readers will also be directed on where to go to download the operating system installation images for creating virtual machines. Finally, some software packages and/or updates that are core to the lab environment will require account registration on a few websites in order to download them. Students will be linked to those websites to register an account.

Be aware that as we mentioned in the previous section, the websites and the links provided in this book are subject to change. Websites get redesigned all the time, and the links provided in this book may become outdated. Sometimes, this may require readers to navigate to the software company's main website or utilize a search engine to find the new URL for downloading the recommended software. Don't panic if the download link is dead. Stay calm, and adapt as needed.

"What in the world is an SCP? I have no idea what any of these acronyms mean!"

This next section is just a list of software to download and install. Included are very brief explanations of what the software does. Hopefully, this will provide understanding in why these applications are needed. Some might already be familiar with these tools and applications, while others may not. Don't panic if you have no idea what an SSH or SCP is, or what two-factor authentication is. Everything will be explained as it relates to the lab, and guidance will be provided on how to use the tools as necessary.

It is also advised that, even if you prefer to use another tool that serves the same purpose as the software listed, that readers download and install the tools specifically listed here. Doing so will make it much easier for you to follow the instructions, instead of having to map the functions in each of these tools to the software you prefer to use. Start with the basics, use the tools listed here, then once your baseline is established, experiment with using different tools or preferred software. Crawl before you walk, walk before you run, run before you soar.

1.5.1 Windows Software Recommendations

- mRemoteNG – a tabbed front-end for a variety of remote access protocols
 - <https://mremoteng.org/download>
- WinSCP – a windows client application for the SCP protocol
 - <https://winscp.net/eng/download.php>
- 7-zip (64-bit, exe) – an open-source application that can handle a variety of compressed file formats
 - <https://www.7-zip.org/download.html>
- puttygen (64-bit) – a windows application for generating SSH keys
 - <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Notepad++ (64-bit) – a powerful text editor
 - <https://notepad-plus-plus.org/download/>
- KeePassXC (64-bit) – a multiplatform password manager
 - <https://keepassxc.org/download/>

1.5.2 MacOS Software Recommendations

• Command-line (CLI) applications (Pre-installed):

- vi/vim
 - A very old and beloved text editor
- ssh
 - secure connectivity to remote systems
- scp
 - Transfers files to/from a remote system utilizing the SSH protocol
- ssh-keygen
 - Generates a public/private key pair. Enables passwordless login, OR basic two-factor authentication for SSH and SCP
- ssh-copy-id
 - a pain-free method of copying SSH public keys to a remote SSH server to enable key-based authentication for SSH and SCP
- zip/unzip, gzip/gunzip
 - utilities that decompress .zip and .gz files
- alias
 - command-line tool to shortcut entire commands plus their arguments to another name

• Third-party Tools (download and install these):

- iTerm2 – a better replacement for the MacOS terminal application
 - <https://www.iterm2.com/downloads.html>
- BBEdit – powerful text editor
 - <https://www.barebones.com/products/bbedit/download.html>
 - **Note:** you do **NOT** need to pay for the licensed features
- KeePassXC – multiplatform password manager
 - <https://keepassxc.org/download/>

1.5.3 Linux Software Recommendations

- Command-line (CLI) applications (Pre-installed):

- `vi/vim`
 - A very old and beloved text editor
- `ssh`
 - secure connectivity to remote systems
- `scp`
 - Transfers files to/from a remote system utilizing the SSH protocol
- `ssh-keygen`
 - Generates a public/private key pair. Enables passwordless login, OR basic two-factor authentication for SSH and SCP
- `ssh-copy-id`
 - a pain-free method of copying SSH public keys to a remote SSH server to enable key-based authentication for SSH and SCP
- `zip/unzip, gzip/gunzip`
 - utilities that decompress .zip and .gz files
- `alias`
 - command-line tool to shortcut entire commands plus their arguments to another name

- Additional Applications (graphical and/or third party)

- Graphical text editor
 - Common editors: `gedit`, `kwwrite`, `leafpad`, `sublime`, `atom`, etc.
- Terminal application
 - Common terminal apps: `Konsole`, `Terminal`, `Gnome-Terminal`
 - Special Mention: `Gnome Terminator`
 - <https://gnometerminator.blogspot.com>
- A window manager
 - `Gnome`, `XFCE`, `Flux`, `KDE`, `Cinnamon`, etc.
- `KeepassXC` – multiplatform password manager
 - <https://keepassxc.org/download/>
 - Not sure how to install on your distro? download the `AppImage`

- Linux Kernel Headers

- See [section 1.7](#) (pp. 28-32)

1.5.4 Operating System Installation Images

- Ubuntu Server – A full-featured, well-documented Linux distribution that will serve as the backbone our lab environment. **This book uses version 20.04. Always use the latest LTS release.**
 - <https://releases.ubuntu.com/>
 - Select the latest LTS release
 - Please download the 'Server install image' ISO
- pfSense – An open-source firewall/router software distribution based on FreeBSD. This software will provide core network services, segmentation, and security for our lab.
 - <https://www.pfsense.org/download>
 - Select 'AMD64 (64-bit)' from the Architecture drop-down, 'CD Image (ISO) Installer' from the Installer drop-down, select the closest physical location from the Mirror drop-down, then click the Download button.
- Kali Linux – The only Linux distribution that allows you to hack the Pentagon, join Anonymous, and Try Harder™. Kali is a distribution built with network penetration testers in mind, bristling with network security tools and frameworks.
 - <https://www.kali.org/get-kali/#kali-bare-metal>
 - Download the 'Kali Linux 64-bit (Installer)' ISO
- Metasploitable 2 – A very old, and very intentionally vulnerable pre-built Linux virtual machine. Along with Kali, we will be using metasploitable to test out network connectivity and security for the lab.
 - <https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>
 - Click the "Download Latest Version metasploitable-linux-2.0.0.zip" button to download the virtual machine image

1.5.5 Register accounts on these websites

- <https://snort.org> - Snort is powerful open-source network inspection software, and one of two choices available for our lab environment
 - Readers will need an "oink code" to install Snort later on.
 - If you are unsure whether or not you want to use Snort or Suricata (another powerful open-source network inspection suite), register an account anyway. It's free and you have nothing to lose.
- **Account registration link:**
https://snort.org/users/sign_up
- <https://www.splunk.com> – Splunk is powerful log management software and will be a major component of our lab.
 - An account is required to download the components we will need later.
 - **Account registration link:**
https://www.splunk.com/page/sign_up?
- <https://www.vmware.com> – VMware provides three out of the five hypervisors this book covers.
 - A My VMware account is required for VMware Workstation, Fusion, or ESXi.
 - **Account registration link:**
<https://my.vmware.com/web/vmware/registration>

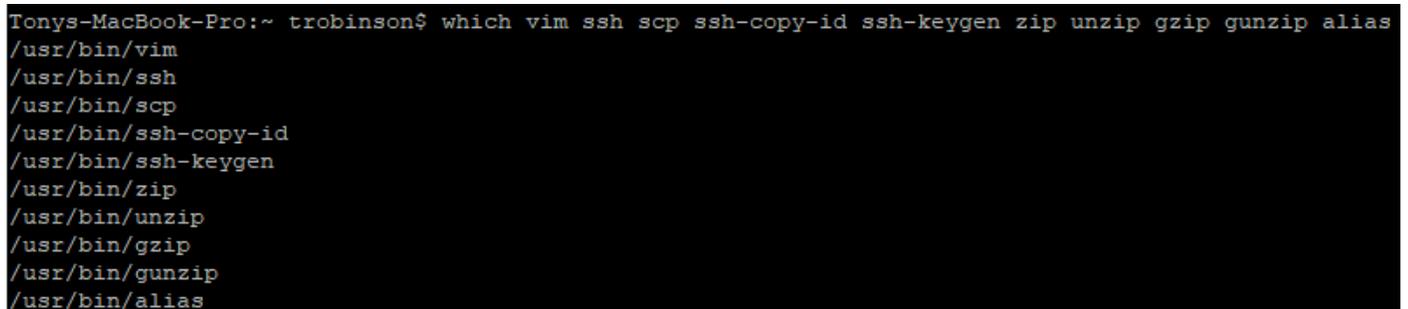
1.6 Linux users, MacOS users and the which command

The command line applications for Linux and MacOS users listed above reads in parenthesis, "pre-installed". Most distributions of Linux, and practically every version of MacOS should have these command-line tools installed already. The quickest way to confirm whether or not the system has these tools installed is through a command line utility called `which`. This command checks to see if an application exists in a list of various places (this is controlled by something called the `PATH` variable). In order to see if these tools are installed on your system, on either Linux or MacOS, open a terminal session, and run:

```
which vim ssh scp ssh-copy-id ssh-keygen zip unzip gzip gunzip alias
```

You should get some output similar to:

```
/usr/bin/vim
/usr/bin/ssh
/usr/bin/scp
/usr/bin/ssh-copy-id
/usr/bin/ssh-keygen
/usr/bin/zip
/usr/bin/unzip
/usr/bin/gzip
/usr/bin/gunzip
/usr/bin/alias
```

A terminal window screenshot showing the command 'which vim ssh scp ssh-copy-id ssh-keygen zip unzip gzip gunzip alias' being executed. The output is a list of paths: /usr/bin/vim, /usr/bin/ssh, /usr/bin/scp, /usr/bin/ssh-copy-id, /usr/bin/ssh-keygen, /usr/bin/zip, /usr/bin/unzip, /usr/bin/gzip, /usr/bin/gunzip, and /usr/bin/alias. The terminal prompt is 'Tonys-MacBook-Pro:~ trobinson\$' and the background is black with white text.

```
Tonys-MacBook-Pro:~ trobinson$ which vim ssh scp ssh-copy-id ssh-keygen zip unzip gzip gunzip alias
/usr/bin/vim
/usr/bin/ssh
/usr/bin/scp
/usr/bin/ssh-copy-id
/usr/bin/ssh-keygen
/usr/bin/zip
/usr/bin/unzip
/usr/bin/gzip
/usr/bin/gunzip
/usr/bin/alias
```

1-1: Output from the `which` command on MacOS. The output should pretty similar on most Linux distributions.

Some of the paths displayed may be different depending on what Linux distribution and/or version of MacOS students are running. So long as the output that shows the commands are installed, that is all that matters.

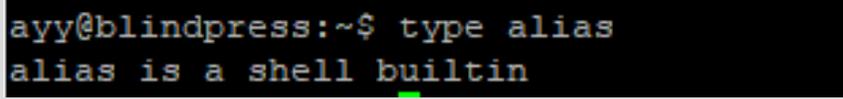
alias is a shell built-in

Linux users: be aware that the `alias` command is considered a shell built-in (that is, something built into the functionality of the command-line software), and that you probably will NOT get an entry for `alias` when running the `which` command. Here is an alternate command you can run on Linux (or MacOS) to confirm that `alias` is present:

```
type alias
```

You should get the output:

```
alias is a shell builtin
```



```
ayy@blindpress:~$ type alias
alias is a shell builtin
```

1-2: the `type` command can be used to check if the `alias` command is available on both Linux and/or MacOS. It may not show up with the `which` command because it is considered a special command called a shell built-in.

1.7 Linux Users and Kernel Headers

One of the software installation requirements listed above for the Linux users was Linux kernel headers. These are source code files for the version of the Linux kernel – the software powering your favorite Linux distribution. Some of the virtualization software this book covers for Linux (Oracle VirtualBox, and VMware Workstation Pro) require it. If students aren't sure what virtualization software they are interested in yet, they may come back here to install the kernel headers when ready. Students will be reminded to have the headers installed when it is necessary.

One of the greatest strengths that Linux provides as an operating system, is that of endless freedom to customize software however you see fit. However, sometimes this strength can be something of a slight annoyance, because every distribution has a different name for referring to the same applications, services, tools and programs. To say nothing of how different Linux distributions manage their software. Some rely on package managers (applications that handle downloading and installing software and their dependencies -- the files that software requires to function), while others make users acquire the source code for the software, manually acquire any dependencies, and compile it themselves. This means that every distribution has some slightly different way of naming and acquiring the kernel headers.

This section will provide users with guidance on how to install the kernel headers package for some of the most commonly used Linux distributions and package managers: Debian-based distributions (Including Ubuntu) using the `apt` package manager, and Redhat-based distributions

(Including CentOS) using the yum, or dnf package managers. Unfortunately, if you are using a more exotic Linux distribution as your desktop operating system, I have to assume that you have the technical aptitude to troubleshoot and refer to that distro's documentation for finding the kernel headers on your own, otherwise we would be here all day, as there are more Linux distros than there are grains of sand in the desert – with more showing up every day.

Please be aware that following these instructions will require accessing and running terminal commands. If students are not comfortable doing so, Chapter 2 will provide a complete list of training resources, including some on how to better navigate the Linux command line. Otherwise, these instructions are as good a place to start as any! Remember that while trying is the first step towards failure, that failure can sometimes be the best teacher.

1.7.1 How to Acquire Kernel Headers for Ubuntu/Debian-based Distributions

Open a terminal application, and run the following commands:

```
sudo su -  
apt-get update  
apt-get -y install linux-headers-generic  
exit
```

The commands above use the `sudo` command to grant a terminal session as the `root` user (if readers are already the `root` user, skip this command). The next two commands update the `apt-get` package manager with the latest listings, then request the `linux-headers-generic` package. Finally, `exit` closes our session as the `root` user.

Kernel headers note for Debian Users

The `linux-headers-generic` package is specific to Ubuntu. Debian users (and Kali users, if you're brave enough to be using Kali as a daily driver operating system) will want to replace `linux-headers-generic` with `linux-headers-amd64` instead.

1.7.2 How to Acquire Kernel Headers for Redhat Enterprise/CentOS-based Distributions

Open a terminal application and run the following commands:

```
sudo su -  
dnf install kernel-devel  
exit
```

Just like with Ubuntu/Debian, root access is required (again, `sudo su -` can be skipped if students are already logged in as root). Then, the `dnf` package manager is used to request the `kernel-devel` package. Finally, `exit` closes the root terminal session.

Note: If you happen to be running CentOS/RHEL 7 or older, you'll need to use the `yum` package manager specifically to install the `kernel-devel` package:

```
sudo su -  
yum install kernel-devel  
exit
```

```
[ayy@localhost ~]$ sudo su -
[sudo] password for ayy:
[root@localhost ~]# dnf install kernel-devel
Last metadata expiration check: 0:21:30 ago on Mon 07 Sep 2020 05:00:24 PM EDT.
Dependencies resolved.
=====
Package                Architecture Version                Repository              Size
=====
Installing:
kernel-devel           x86_64                4.18.0-193.14.2.el8_2  BaseOS                  15 M
=====
Transaction Summary
=====
Install 1 Package

Total download size: 15 M
Installed size: 47 M
Is this ok [y/N]: y
```

```
root@ayy: ~
ayy@ayy:~$ sudo su -
[sudo] password for ayy:
root@ayy:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
root@ayy:~# apt-get install linux-headers-generic
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  linux-headers-generic
0 upgraded, 1 newly installed, 0 to remove and 95 not upgraded.
Need to get 2,584 B of archives.
After this operation, 17.4 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 linux-headers-g
generic amd64 5.4.0.31.36 [2,584 B]
Fetched 2,584 B in 0s (19.3 kB/s)
Selecting previously unselected package linux-headers-generic.
(Reading database ... 156951 files and directories currently installed.)
Preparing to unpack .../linux-headers-generic_5.4.0.31.36_amd64.deb ...
Unpacking linux-headers-generic (5.4.0.31.36) ...
Setting up linux-headers-generic (5.4.0.31.36) ...
root@ayy:~# exit
```

1-3: The top-most image depicts the install process for the `kernel-devel` package using the `dnf` package manager for Redhat-derived Linux distributions. If students are using a Redhat-based Linux distribution and the `dnf` package manager is not available, try using the `yum` package manager instead.

The next image depicts the install process for the `linux-headers-generic` package on Ubuntu. Be aware that some Debian-based distros name the software package `linux-headers-amd64`, instead. Remember to exit the terminal session after running tasks that require root access.

sudo voodoo

`sudo` is a command that allows a normal user (provided they have the correct access) to run a single command as the `root` user by entering their login password. Some regard using `sudo` (as opposed to logging in as the `root` user account) as best practice. In fact, the Ubuntu Linux distribution does not allow users to log in as the `root` user (by default... We'll talk more about this *much* later). You can still do some interesting things that allow you to run multiple commands as `root` (like, say, `sudo su -`, or `sudo /bin/bash`) but, overall, the `sudo` method is a little bit safer. This is because logging in as `root` (or using a `root` terminal session) grants the ability to run multiple commands as the `root` user.

The `root` user has complete authority to tell the system to do practically anything – even if the user behind the account has no idea what those commands will actually do. The potential to shoot yourself in the foot by destroying data, or damaging your Linux installation is much higher. Still, others (like me) find using `sudo` to be very tedious if you have a lot of work and customizations to do, and they all require `root` access. This lab environment is your opportunity to experiment, learn best practice or be lazy (like me) if that's what you want. I want you as students to have the freedom and knowledge to make these sorts of choices yourself, so here is a slightly faster and safer alternative to the commands provided above, that still accomplishes the same task of installing the kernel headers:

Ubuntu/Debian-based distros:

```
sudo apt-get update
sudo apt-get install linux-headers-generic
```

Remember to replace `linux-headers-generic` with `linux-headers-amd64` if necessary (e.g., Kali, Debian, etc.)

Redhat/CentOS-based distros:

```
sudo dnf install kernel-devel
```

Or, if `dnf` is not present, try `yum` instead:

```
sudo yum install kernel-devel
```

The main difference is that we prefaced `sudo` in front of the package manager commands. That way, we only used `root` access to run the commands that required it, then immediately relinquished that access after the command completes. In this case, since it's just a single task we were performing, it's pretty fast, but imagine if you had a dozen or more commands that all require `root` access. You might see where `sudo su -` would be pretty enticing, right?

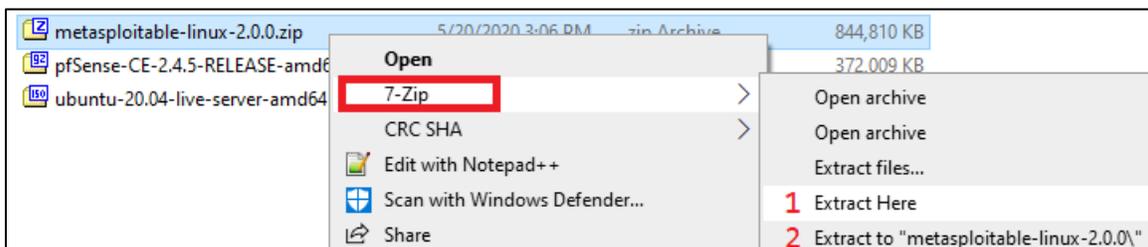
1.8 Using Compression Tools

In [section 1.5.4](#) (p. 26), readers were provided with a list of ISOs and pre-built virtual machines to download. Some of these files are compressed by default, using `gzip` (`pfSense-CE-x.x.x-RELEASE-amd64.iso.gz`) or `zip` compression (`metasploitable-linux-2.0.0.zip`). In this section, readers who are not familiar with compression tools will be provided a brief tutorial on how to decompress both the `pfSense` and `metasploitable 2` files for use in later chapters.

Some of these instructions (for the Linux users especially) may require interacting with the command line. As mentioned in the previous section, don't be afraid to jump in feet first. This lab environment is a place for learning, and now is as good a time as any to make mistakes in order to further that goal. If students are still not comfortable performing these tasks yet, check out chapter 2 for links to training in order to get more familiar with navigating desktop operating systems, and/or navigating the command line.

1.8.1 7-Zip on Windows

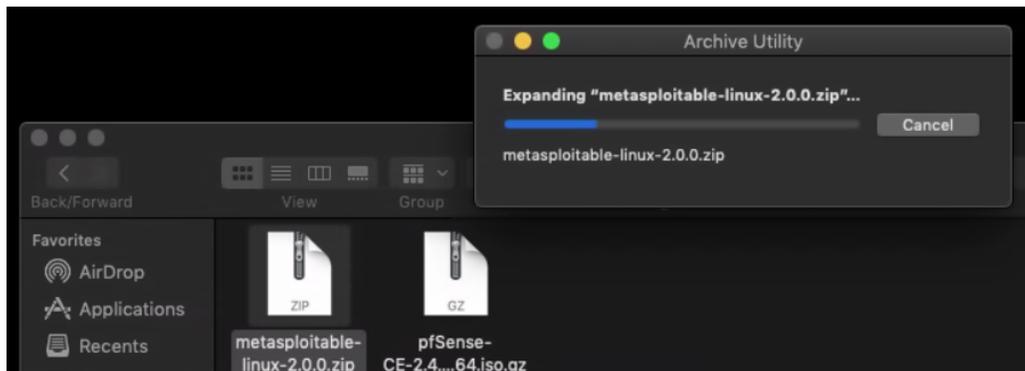
7-Zip is incredibly easy to use, and can compress or decompress files using a variety of different file compression formats and features. We will be using 7-Zip to handle both the `pfSense .gz` file, and `metasploitable 2 .zip` file. By default, the utility integrates itself into Windows Explorer as a right-click context menu option. Right-click on the file to decompress, select *7-Zip* from the context menu, then select *Extract Here* to extract the compressed file to the current directory. Alternatively, there is an option that will create a folder named after the compressed file, and drop the contents of compressed file to that directory (for example, *Extract to 'metasploitable-linux-2.0.0/'* as an option for `metasploitable-linux-2.0.0.zip`) in the current folder the compressed file is located in.



1-4: Right-click `metasploitable-linux-2.0.0.zip`, and select *7-Zip* from the context menu. From here, users can select *Extract Here* (1) to decompress and extract the contents of the zip file here (A folder named `Metasploitable2-Linux` with files in it), or *Extract to "metasploitable-linux-2.0.0\'"* (2) to create a folder with that name, decompress the zip file, and move its contents (The directory `Metasploitable2-Linux` and its files) to that new folder. This process can be repeated with the gzipped `pfSense` ISO (`pfSense-CE-x.x.x-RELEASE-amd64.iso.gz`) as well.

1.8.2 Finder on MacOS

Finder is the built-in file browser for MacOS. One of the really neat features Finder has by default, is the ability to understand different types of file compression (including .zip and .gz files), and decompress them for you instantly using its Archive Utility. It's as easy as double-clicking on the file you wish to decompress, and letting Finder do all the heavy lifting. Alternatively, for fans of the command line (or doing things the hard way), continue to section 1.8.3 to learn how to use the `zip/unzip`, and `gzip/gunzip` commands.



1-5: Use Finder to locate the compressed file (in this instance, `metasploitable-linux-2.0.0.zip`), and double-click on it. Archive Utility will do the rest. Repeat for `pfSense-CE-x.x.x-RELEASE-amd64.iso.gz`. If for some reason this does not function correctly, check out section 1.8.3 to learn how to use command-line decompression tools.

1.8.3 `zip/unzip` and `gzip/gunzip` on Linux (and MacOS)

Some Linux distributions and window managers may have access to utilities like File Roller that will allow users to decompress files from their file manager (sort of like MacOS Finder's Archive Utilities). However, since there are as many Linux distributions as there are stars in the sky, it cannot be assumed that all users will have access to those convenient utilities. Command-line tools are universal – practically every Linux distro will have `zip/unzip` and `gzip/gunzip` available for handling zipped or gzipped files. In the cases where these tools aren't present, they can easily be acquired using the Linux distro's package manager.

Open a terminal application to access the command line on your Linux distro (MacOS users: Open the *iTerm* or *iTerm2* applications). Navigate to the directory the `metasploitable-linux-2.0.0.zip` and/or `pfSense-CE-x.x.x-RELEASE-amd64.iso.gz` files are located by using the `cd` command. For example, if they are located in `/home/ayylmao/Downloads`, run:

```
cd /home/ayylmao/Downloads
```

Alternatively, assuming the current user is ayy1mao, one could use the '~/' shortcut to refer to the current user's home directory. That command would look like:

```
cd ~/Downloads
```

Once in the directory where the compressed files are located (the `ls` command can confirm this), Students will need to decompress the files. For `metasploitable-linux-2.0.0.zip`, run:

```
unzip metasploitable-linux-2.0.0.zip
```

For `pfSense-CE-x.x.x-RELEASE-amd64.iso.gz`, run:

```
gunzip pfSense-CE-x.x.x-RELEASE-amd64.iso.gz
```

Note that these filenames are a little bit on the long side, and have to be exactly right for the `unzip` or `gunzip` commands to decompress the correct files, but a nice trick in most Linux/Unix command line environments (including MacOS) is a feature called tab completion. If you start typing in the name of a file and hit the Tab key, the command-line interpreter will try to complete the name of the file for you, making this task much easier. Also note that **by default, gunzip will delete the original compressed file after decompressing it.** You'll need to run `gunzip -k [filename.gz]` to keep the original file.

```
ayy@ayy:~$ cd ~/Downloads
ayy@ayy:~/Downloads$ ls
metasploitable-linux-2.0.0.zip  pfSense-CE-2.4.5-RELEASE-amd64.iso.gz
ayy@ayy:~/Downloads$ unzip metasploitable-linux-2.0.0.zip
Archive: metasploitable-linux-2.0.0.zip
  creating: Metasploitable2-Linux/
  inflating: Metasploitable2-Linux/Metasploitable.nvram
  inflating: Metasploitable2-Linux/Metasploitable.vmdk
  extracting: Metasploitable2-Linux/Metasploitable.vmsd
  inflating: Metasploitable2-Linux/Metasploitable.vmx
  inflating: Metasploitable2-Linux/Metasploitable.vmx
ayy@ayy:~/Downloads$ gunzip pfSense-CE-2.4.5-RELEASE-amd64.iso.gz
ayy@ayy:~/Downloads$ ls
Metasploitable2-Linux      pfSense-CE-2.4.5-RELEASE-amd64.iso
metasploitable-linux-2.0.0.zip
```

1-5: Navigate to the directory where the file(s) to decompress are located using the `cd` command. Run `ls` to confirm the files are present in the directory. From there, it's a matter of running `unzip` against `.zip` files, or `gunzip` against `.gz` files. Notice in the second `ls` command that the `pfSense-CE-x.x.x-RELEASE-amd64.iso.gz` file is no longer there, because `gunzip` was ran without the `-k` option.

Chapter 2 Patch Notes

- Reorganized the list of recommended skills
- Explained that not having a skill (or skills) listed doesn't mean you can't follow along, it'll just make things a little easier to understand
- Provided a better, updated list of training resources for those who want to learn more with special thanks to Julia "b0rk" Evans, Elan "DFIRDiva" Wright, and Dennis Devey.

Chapter 2: Recommended Skills and Knowledge

This book is designed to where just about anyone should be able to pick it up, and successfully create their first virtual machine environment. However, with knowledge of a few basic IT disciplines, the instructions and design concepts will become easier to understand. This chapter contains a list of questions and concepts. The more of these questions students can answer, the easier the lessons taught in this book will be to grasp.

For students that are having a hard time with these questions, who just want access to extra resources to fill in any gaps, I've also provided a list of external resources to free, or relatively affordable training.

2.1 TCP/IP Networking

- What is an IP address?
- What is a default gateway?
- Can students configure an IP address, netmask, default gateway, and DNS servers on Windows, Linux, and/or MacOS?
- How familiar are students with configuration files and/or utilities used to retrieve network information on Windows, Linux, and/or MacOS?
 - ipconfig /all, ncpa.cpl, ifconfig, ip [address, route, link], route, netstat, ss, /etc/resolv.conf, /etc/hosts, etc.
- What is an RFC1918 address?
- What are the RFC1918 address ranges?
- What is the OSI network model? What about the TCP/IP network model?
- How do stateful firewalls operate? What makes them stateful?
 - what criteria are used to create firewall rules?
 - IPv4/IPv6 source/destination addresses
 - transport protocol (tcp/udp/icmp)
 - source/destination port/icmp code
- What ports are used for common network protocols?
 - What are the port numbers associated with these protocols?
 - FTP
 - HTTP
 - HTTPS
 - NTP
 - SSH/SCP
 - DNS
 - Are the services above typically TCP or UDP?

2.2 Navigating Operating Systems, and their Installation Procedures

- Familiarity with installing Linux/Unix/Windows operating systems
- What an ISO image?
- What is the process for booting a computer (or virtual machine) from an ISO image?
- Familiarity with the Unix/Linux/Windows command line
 - The Linux/Unix shell environment (`bash/zsh`)
 - The Windows command-line (`cmd.exe, powershell.exe`)
- What are some common network troubleshooting utilities used to test connectivity?
 - `ping, wget, curl, etc.`
- What are some common Unix/Linux command line text editors? Are you comfortable using them?
 - `vim, nano, emacs, etc.`
- Are students familiar with the `ssh, ssh-keygen, and/or scp` commands on Linux and MacOS?
- Are students familiar with PuTTY, PuTTYgen and/or WinSCP on Windows?

2.3 Recommended Training Resources

For those who wish to learn more, here is a list of recommended books and affordable training resources:

- **Practically anything Published by No Starch Press.** No Starch is well-regarded in the information security community as being a publisher of quality computer books that serve as excellent reference material. Here are a few books that I recommend:
 - *Network Know-How – An Essential Guide for the Accidental Admin*
 - *TCP/IP Guide – A comprehensive, Illustrated Internet Protocols Reference*
 - *Practical Packet Analysis – Using Wireshark to Solve Real-World Network Problems*
 - *How Linux Works – What Every Superuser Should Know*
 - *The Linux Command Line – A complete Introduction*
 - *Linux Basics for Hackers – Getting Started with Network, Scripting, and Security in Kali*

Note: Keep an eye on Humble Bundle (<https://www.humblebundle.com>). No Starch has a tendency to partner with them, and will provide digital (PDF) copies of their books bundled at a vastly discounted price.

- **Wizard Zines (<https://wizardzines.com>)** – Julia Evans creates short, notebook styled PDFs (called zines), filled with knowledge about system and network troubleshooting commands, and how they can be used. There is a wide variety of content available.

- **Alison (<https://alison.com>)** – Alison is a website that is jam-packed with video-based training for a wide spectrum of topics. There are a number of courses that grant a certificate upon completion, as well as some course collections that grant a diploma for completing the entire series.
- **Sans Cyber Aces (<https://www.cyberaces.org/>)** – Sans is a world-renowned provider of high quality (but extremely expensive) IT Security training. Cyber Aces is their effort to provide access to free training material to any who are interested in transitioning to IT and/or IT Security. There is a large selection of learning resources available on Windows, Linux, and Networking to help you get up to speed quickly.
- **Coursera (<https://www.coursera.org/>)** – Coursera is another video-based learning website. Not unlike Alison, there is a wide variety of content out there. Most of the material comes directly from universities and several technology companies, and most offer completion certificates as well.
- **Roppers Academy (<https://www.hoppersroppers.org/training.html>)** – Roppers Academy is a collection of training courses organized by Dennis Devey. The training has an information security focus, with the express goal of allowing "aspiring cyber security professionals to learn and grow". While some of the material is still a work in progress, the Introduction to Computing Fundamentals curriculum covers a huge variety of different topics, and provides a very strong starting point. The course is very hands-on with the instructor telling you what they would like you to accomplish with a lesson, providing links to resources to help you meet that goal, and providing a Slack channel for students to seek help.
- **Practically any vi/vim text editor tutorial** – Much later in this book, there will be a lot of files that students need to modify or create on Linux virtual machines. The easiest way to do this will be through learning a command-line text editor that is already present by default on most Linux/Unix systems, and that is the `vi` or `vim` text editors. The thing is, in spite of being ubiquitous, they aren't exactly user friendly. However, as you get use to the quirks of `vi`, you'll be a pro in no time. There are multitudes of quick reference cards, and tutorials, but Daniel Miessler wrote an exceptional tutorial for `vim` at <https://danielmiessler.com/study/vim>. If students are already familiar with another command-line text editor (e.g., `nano`, `ed`, `pico`, `emacs`, etc.) that they are more than welcome to use those text editors instead.
- **DFIR Diva's training resource list** – Elan "DFIRDiva" Wright has organized a huge collection of IT training resources covering a wide variety of subjects. Elan has a collection of training resources that are paid trainings (<https://dfirdiva.com/training>) as well as a collection of free training materials, that are broken up by category (<https://freetraining.dfirdiva.com/>).

Chapter 3 Patch Notes

- Re-worded and re-organized quite a few things to make it flow better.
- Changed the title of the chapter to better reflect the subject matter being discussed
- New screen captures added

Chapter 3: Virtual Machines and Hypervisors

Before creating our lab, it's important to understand the building blocks on which it will be created. This chapter will serve as an introduction to virtualization and hypervisors, explaining what a virtual machine (often referred to as a "VM") is, as well as the differences between bare-metal and hosted hypervisors.

3.1 What is Virtualization?

The easiest way to describe virtualization is to think of it as taking one physical computer and allocating a portion of its resources into a dedicated container in order to host smaller, independent virtual systems using those reserved resources. For example, let's take a computer with 2 CPU cores, 4GB of RAM, and 40GB of disk space. Through the use of a virtualization software (called a "hypervisor"), a portion of those resources could be reserved to create a container. This container is a virtual machine; a virtualized computer.

VMs, for all intents and purposes, are independent computers. Like any physical computer, virtual machines require an operating system in order to boot. Any operating system a user has licensing for can be installed on that VM. The number of virtual machines you can create and host on a single physical computer is limited only by the hardware resources available – RAM, CPU, Disk Space, and Disk I/O.

Virtualization is extremely powerful for enterprises because it allows companies to run more services and do more things with less physical hardware. Virtualization is also great for individuals (e.g., developers, researchers, sysadmins, students, etc.) because it allows for the creation of testing environments full of virtualized systems for doing all sorts of experimentation without requiring a large number of physical computers. Not only that, but most virtualization software supports a feature called snapshotting.

Snapshots allow users to capture the status of a VM at a given point in time, and restore the VM to that state whenever they would like. This means that it's possible to create a known good configuration of a virtual machine, snapshot it, and use that as the VM's baseline. This allows users to perform software test cases, make system configuration changes, and/or even run

malware in the VM with little to no concern. Once testing and analysis is finished, the virtual machine can be reverted back to the state it was in when its snapshot was created.

3.2 What is a Hypervisor?

Now that students know what a VM is, let's talk about hypervisors. A hypervisor is the software that is used to create, manage, and run virtual machines. The hypervisor is responsible for allocating hardware resources (RAM, CPU, disk, etc.), network configuration, addition or removal of other virtual hardware, managing snapshots, and controlling the current operating status of the VMs they manage (e.g., powered on, powered off, paused/suspended, etc.). Generally speaking, there are two types of hypervisors recognized today: hosted and bare-metal hypervisors

3.2.1 Hosted Hypervisors

A hosted hypervisor is virtualization software that runs on top of an operating system already installed on physical hardware. The hypervisor is an application "hosted" by the installed operating system (Windows, Linux, MacOS, etc.), thus the name. Usually some sort of a console, or a graphical interface is installed that allows the user to configure the hypervisor and manage their VMs. Some hosted hypervisors do allow you to enable a web-based interface or other remote access, so that configuration changes to the hypervisor and VMs can be made remotely. Popular hosted hypervisors include VMware Workstation, VMware Fusion, Oracle VirtualBox, and many others.



3-1: This is the graphical console for the hosted hypervisor, Oracle VirtualBox. This particular installation is running on top of MacOS. There are a host of icons along the top of the window (and various configuration menus) that allow the user modify various aspects of the hypervisor, and create virtual machines to suit their needs.

3.2.2 Bare-metal Hypervisors

Bare-metal hypervisors differ from hosted hypervisors in that they are usually installed directly on to server hardware (In some IT and Sysadmin circles, servers are sometimes referred to as "Big Iron", or "The Bare Metal" – Hence, the name for this type of hypervisor). Bare-metal hypervisors use a very lightweight operating system with just the minimum amount of functionality required to manage the physical hardware itself, and perform its primary functions as a hypervisor. In fact, some bare-metal hypervisors are designed to where they can be installed and booted from removable media – such as SD cards, or USB drives. This leaves the server's internal hard drives and/or solid-state drives completely dedicated to the virtual machines it will be running.

In most cases, the bare-metal hypervisor is installed, provided with a network configuration, then is managed remotely over the network through an API, web interface, or some other application installed on a management workstation. While bare-metal hypervisor installations tend to be extremely lean, they are packed with a lot of advanced features that are focused more towards enterprise users and customers. These are features such as fault tolerance, high availability clustering, advanced virtual networking configurations, etc. Bare-metal hypervisors are heavily utilized in enterprise environments all over the world, so setting up a lab environment to learn how to use them can be very beneficial.

While bare-metal hypervisors are very powerful and feature-packed, there are some downsides. When users install a bare-metal hypervisor, that hypervisor is the operating system for that physical hardware. That hardware is solely dedicated to managing and running virtual machines, and nothing else. That means you'll need at least one other system to serve as a management workstation in order for you to interact with the hypervisor and do anything useful with it. Another downside to bare-metal hypervisors is that some are extremely picky about the hardware they run on. It is very important to do research and ensure that the server make, model, manufacturer and/or hardware is fully compatible with the bare-metal hypervisor you wish to use. Popular bare-metal hypervisors include VMware ESXi, Microsoft Hyper-V, and Proxmox among others.

The screenshot displays the VMware ESXi web interface for the host **RagnorokESXi.hsd1.mi.comcast.net**. The interface includes a top navigation bar with icons for home, refresh, and settings. A central server icon is shown next to the host name. On the right side, resource utilization is displayed with progress bars: CPU (5% used, 21.6 GHz free, 1.1 GHz used, 22.7 GHz capacity), MEMORY (13% used, 110.88 GB free, 17.01 GB used, 127.89 GB capacity), and STORAGE (2% used, 21.36 TB free, 482.9 GB used, 21.83 TB capacity). The main content area shows host details: Version 7.0.0 (Build 15843807), State Normal (not connected to any vCenter Server), and Uptime 17.45 days. A 'Hardware' section is expanded to show a table of system specifications.

Hardware	
Manufacturer	Cisco Systems Inc
Model	CPS-UCSM4-2RU-K9
CPU	12 CPUs x Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz
Memory	127.89 GB

3-2: This illustration is from VMware's ESXi web interface. This is the host information screen and displays hardware information and resource utilization for the server the user is currently logged in to.

What about Containers?

Some of the more technically savvy students out there may have heard about, or may already be using containerization and container technology. What is a container? What are the differences between a container and a virtual machine?

In a nutshell (I am *vastly* oversimplifying this), containers are semi-independent chunks of resources, sort of like virtual machines. They are lightweight and purpose-built to run either a single, or a small collection of applications. Most containers are in the form of an image that has just the bare minimum configurations, libraries and dependencies needed to run the services they are advertised to.

They are sort of, but not really independent from their host operating system. They're based off of technology called *cgroups* and *namespace isolation*. Like all tools and software, they have their advantages and disadvantages. If you would like to learn more, there is an excellent introduction available here:

<https://www.freecodecamp.org/news/demystifying-containers-101-a-deep-dive-into-container-technology-for-beginners-d7b60d8511c1/>

Chapter 4 Patch Notes

- Repositioned Chapters 4 and 5, because it makes more sense to talk about types of virtual networking immediately after explaining what a virtual machine is, and the different types of hypervisors
- Gave a very brief explanation of virtual networking that I feel the first edition was lacking
- Created new network diagrams to help students better visualize how different types of virtual networking operate

Chapter 4: Introduction to Virtual Networks: Hosted vs. Bare-metal Hypervisor Networking

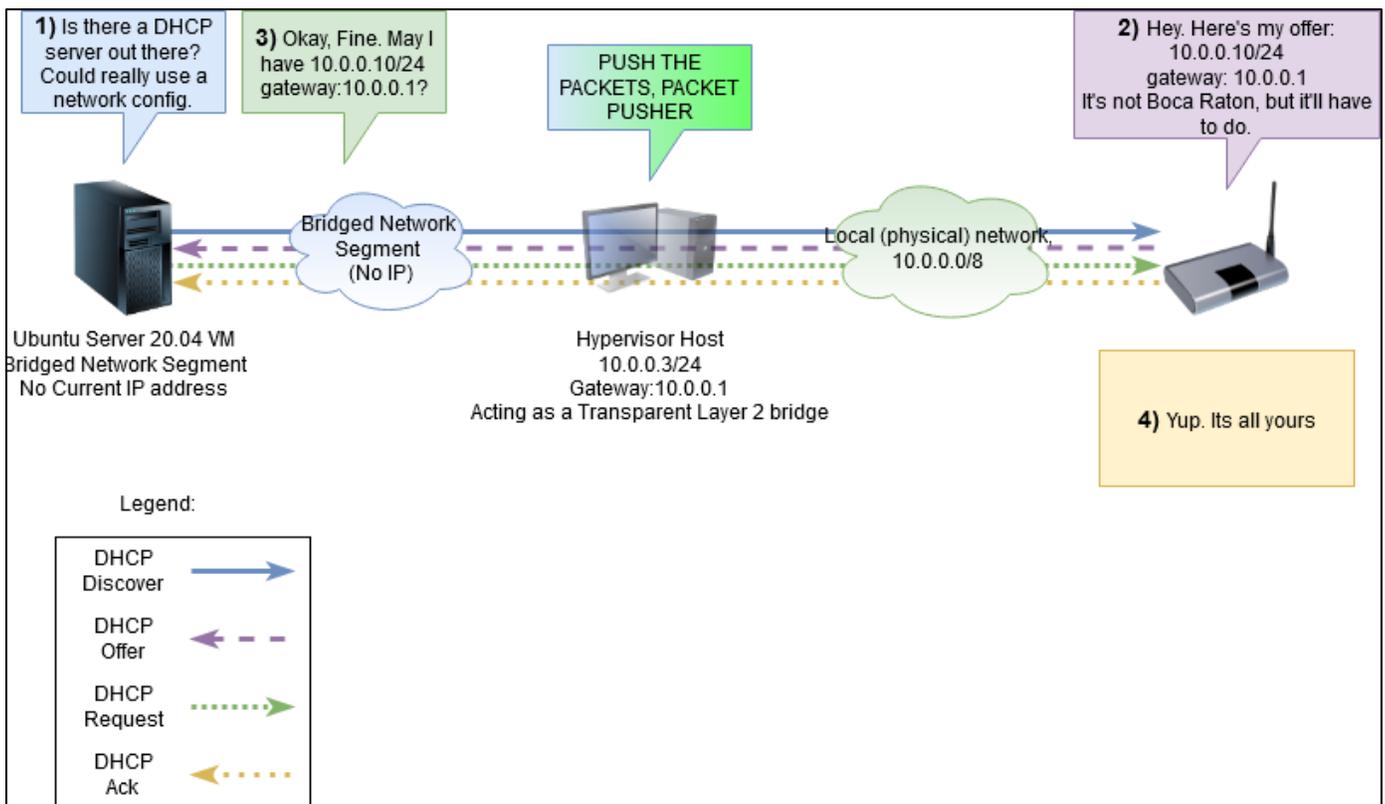
Virtual machines are usually created with some sort of a virtual network card that is attached to a virtual network segment, or a virtual switch port of some sort. The type of virtual network the VM's network card is attached to usually depends on the type of hypervisor being used, and the level of network access the user wants the virtual machine to have. The goal of this chapter is to introduce readers to virtual networking concepts on both hosted, and bare-metal hypervisors. This knowledge is fundamental to understanding the network design of our lab environment, and how to interact with their virtual machines using a variety of network protocols.

4.1 Hosted Hypervisor Networking – Host-Only, Bridged, and NAT Network segments

Virtual Machines on most hosted hypervisors, are provided different types of network access by connecting their network cards to specially labeled network segments: NAT, Bridged, or Host-Only. Some hosted hypervisors are unique and may allow users to create additional, custom network segments to allow for more complex network design but even then, those custom network segments usually fit into one of the three basic types of virtual network segments mentioned. Let's learn more about these network segments and the network access they provide to connected virtual machines.

4.1.1 Bridged Networking

Bridged networking allows virtual machines to interact directly with the same local, physical network the system hosting the hypervisor is connected to. The hypervisor host acts like a network bridge, and just forwards packets between the physical network and the bridged virtual network segment. From the perspective of any other host on the physical network, virtual machines attached to a bridged network segment look like any other system attached to the *actual* physical network – They can request access to network services on that physical network (e.g. request an IP address, subnet and default gateway through DHCP), and will respond to network requests if the virtual machine is hosting a particular service, and the network access is not blocked by a firewall (e.g., if a virtual machine on a bridged segment has a web service installed, such as NGINX or Apache, systems on the physical network will be able to request access the web server hosted that virtual machine).



4-1: This is a network diagram to demonstrate how bridged network segments on a hosted hypervisor work. Here we see an Ubuntu server VM talking to the physical network's DHCP server to get an IP address. The hypervisor host acts as a layer 2 bridge and transparently forwards the DHCP traffic to and from the Ubuntu VM. For all intents and purposes, the DHCP server just sees it as another host on the physical network in need of an IP address.

4.1.2 NAT Networking (and Port Forwarding)

Network Address Translation was originally devised as a way to allow multiple devices on a private network (networks that use private IP addressing – RFC1918 addresses) to "share" public IP addresses for requesting access to services on the internet. NAT enables a network device to act as a connection broker, keep track of which hosts on the private network are making external network requests, make those requests on their behalf, then forward the responses back to the correct private IP addresses. In relation to VMs and virtual networking, the NAT network segment performs a similar function.

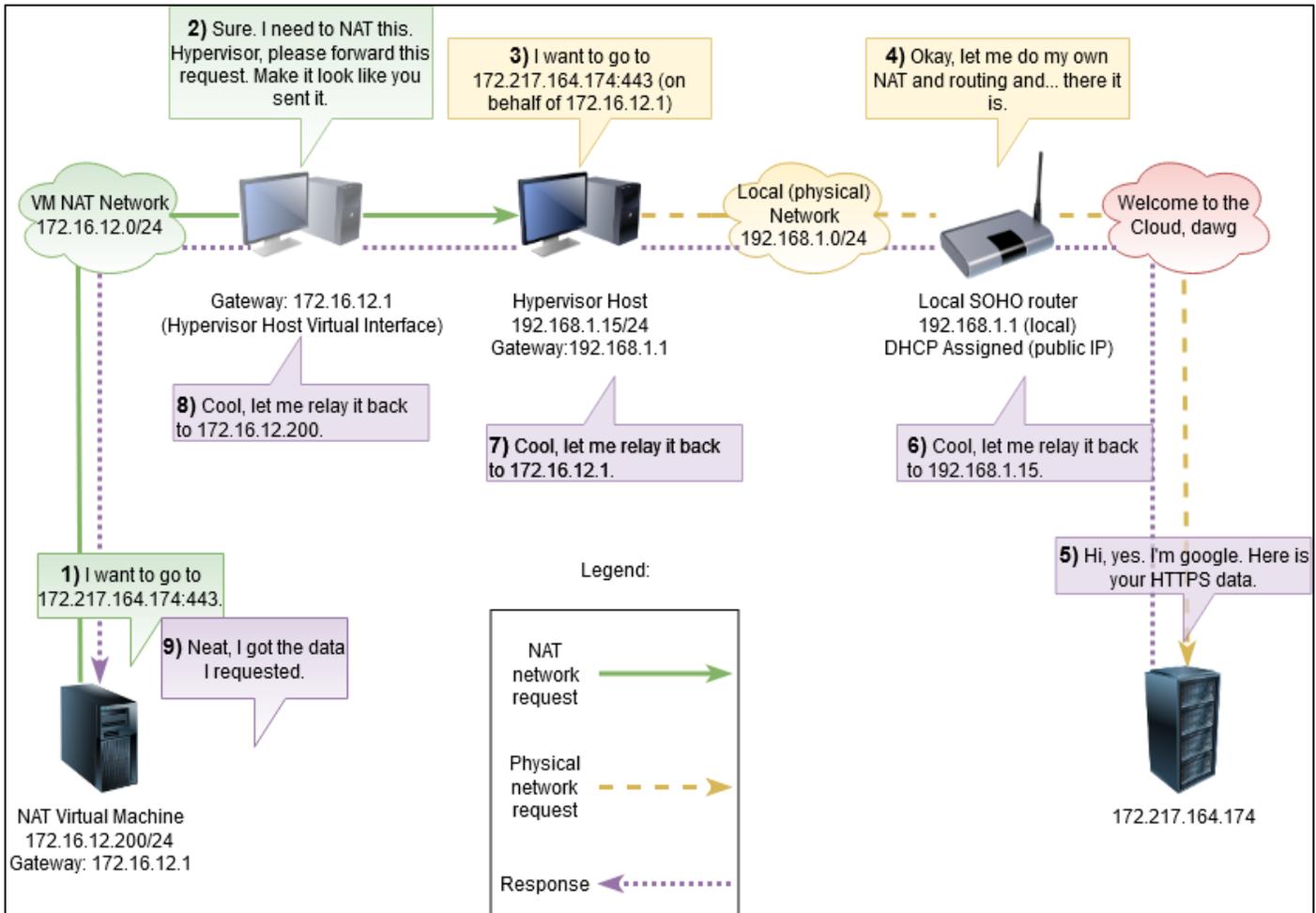
NAT segments have their own network configuration (e.g. IP address range, subnet mask, default gateway, etc.) that is independent of other virtual network segments, and the physical network the hypervisor host is connected to. When virtual machines on a NAT network segment make requests for external network resources, the hypervisor host acts as connection broker and makes those connection requests on behalf of the VMs, then forwards the responses back to the requesting virtual machines.

Port forwarding on the other hand is like NAT, but in reverse. A host from an external network wants to access a service running on a system located on a private network – behind a NAT device with a public IP address. The NAT host can be configured with a port forwarding rule, and when it receives a request to its public IP address on the configured TCP or UDP port, the NAT device can then forward that connection to the system on the private network, on the actual port the service is listening on. The service will respond back to the NAT device, and the NAT device will relay the data back to the external client.

Most hypervisors with a NAT segment have a configuration file buried somewhere in the installation files, or configuration menus that allows you to set up similar port forwarding rules. When your hypervisor host receives a connection to its IP address on a specific TCP or UDP port, it can be configured to relay that connection to an IP address and port on the NAT network segment, and relay the responses back to external clients.

In most cases, I recommend avoiding the use of NAT network segments for virtual machine labs, unless it can't be avoided. For example, some enterprise networks might have NAC (Network Access Control), switchport security, MAC address filtering, or require some form of network authentication to prevent unauthorized devices from accessing the network. This means that bridged virtual machines may be blocked as unauthorized systems. However, if you reconfigure those same virtual machines to use the NAT network segment and services the hypervisor provides, all of the outbound traffic from your virtual machines will appear to be coming from the host system's IP address. This may allow students to work around restrictive network access controls. On the other hand, if they need customize the NAT configuration and/or configure port forwarding to a NAT network segment, it isn't very user friendly. As mentioned above, the

configuration settings tend to be buried on most hypervisors, and on others, the configuration file that controls how the NAT segment operates may need to be edited manually (with a text editor) to configure port forwarding or other customizations.



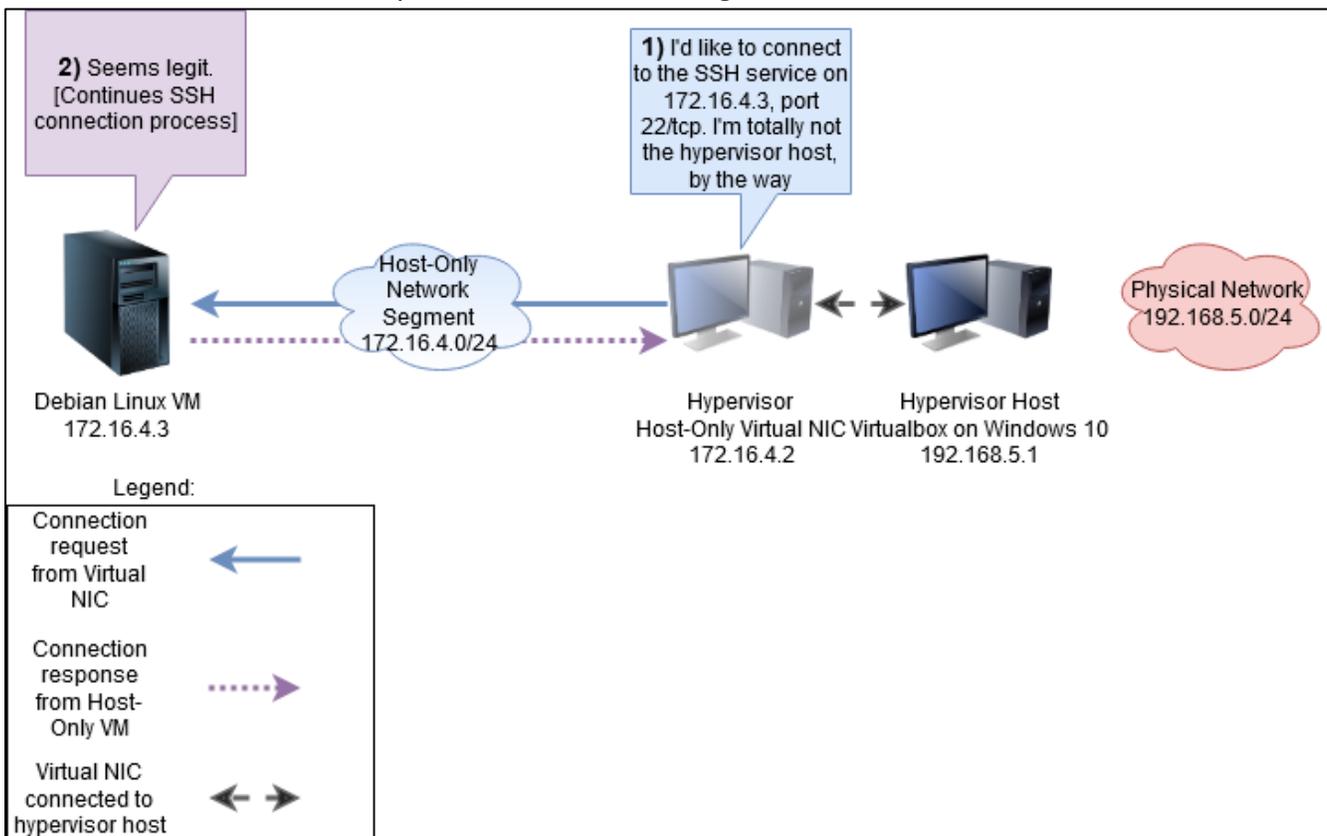
4-2: This a very simplified network diagram of what a network request to google.com looks like from a Virtual Machine attached to the NAT network segment of a hosted hypervisor. Technically this is a Double NAT (since you have to NAT again at step 4), but I won't say anything if you don't.

4.1.3 Host-Only Networking

Host-only virtual networks are network segments that, by definition, do not offer virtual machines connected to them any sort of external network connectivity. Virtual Machines on these network segments can only communicate with hosts on the same host-only network segment, and/or the hypervisor itself. This type of network segment is typically used to isolate virtual machines and limit their access.

4.1.3.4 Virtual Network Adapters

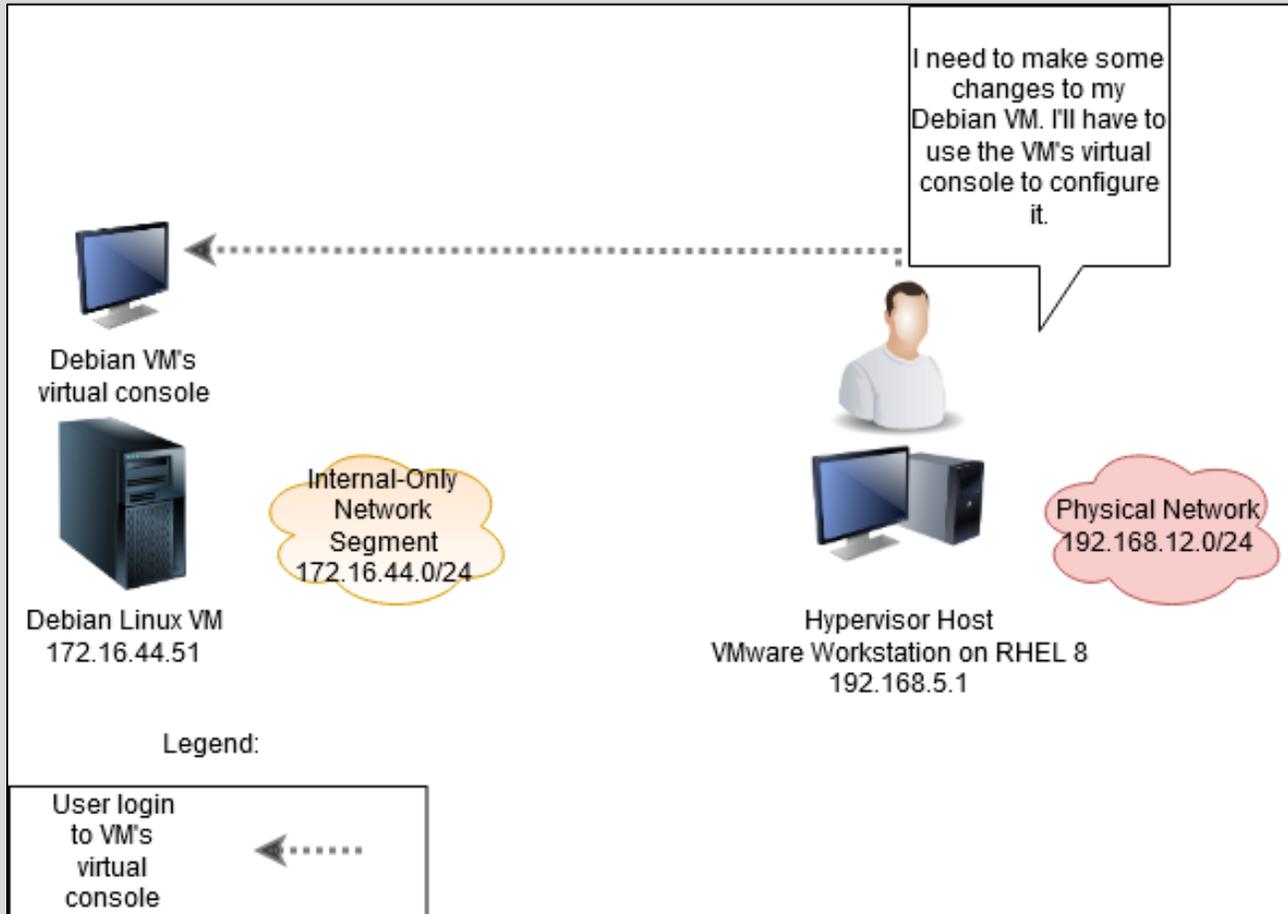
Most hosted hypervisors (and some of the more interesting bare-metal hypervisors) allow the user to create virtual network cards on the hypervisor's host operating system. These virtual network cards allow the host operating system to directly access the network segment they are attached to as though the hypervisor host was another virtual machine on that local network. This is useful for enabling network access to host-only virtual machines to and from the hypervisor host for transferring files, or to allow the hypervisor host to directly access network services hosted on virtual machines in that network segment. Typically, virtual network adapters are connected to host-only and/or NAT network segments



4-3: This illustration demonstrates how host-only network segments work, as well as demonstrating how virtual network cards enable the hypervisor host to directly access virtual network segments. The virtual network card (172.16.4.2) on the hypervisor host is treated just like a physical network card, and can be configured with an IP address to enable network communication with VMs on the network segments they are attached.

Internal-Only Network Segments: A Unique Type of Host-Only Network

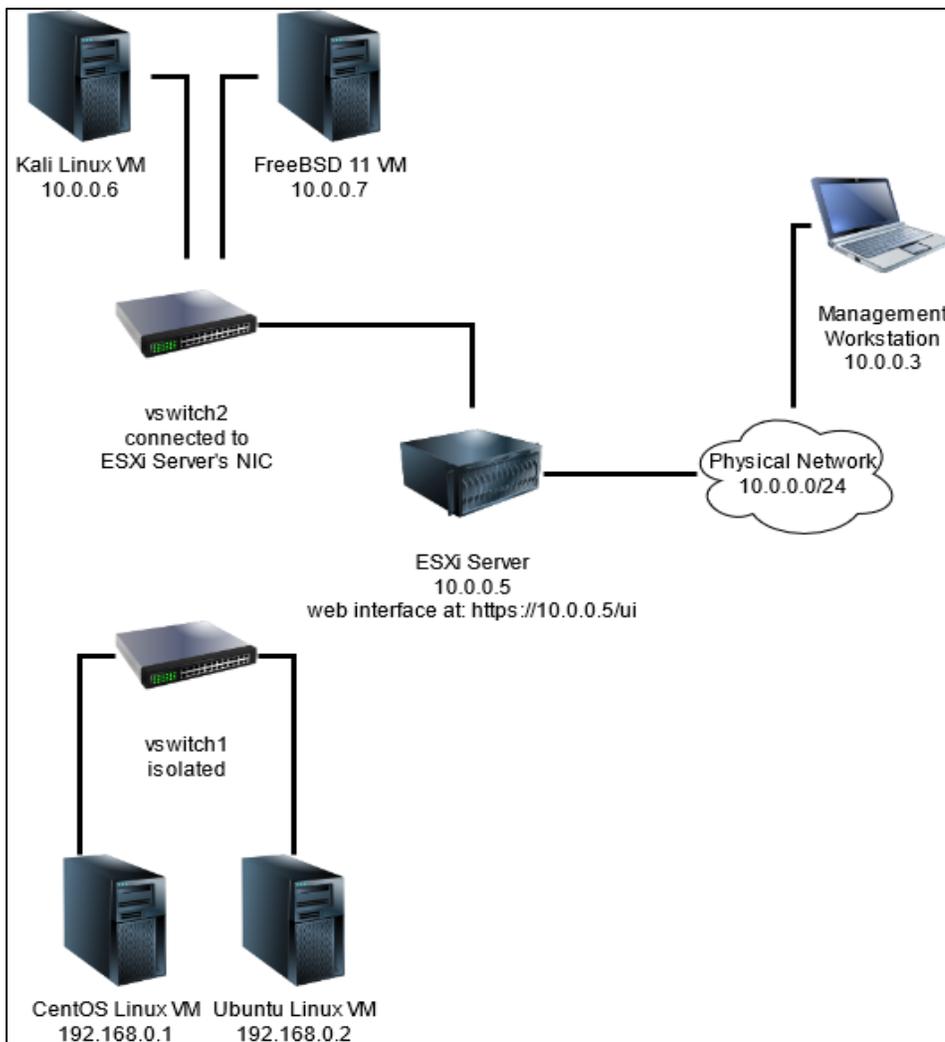
Some hypervisors allow you to disable the creation of these virtual network cards, or create special "internal only" network segments where the only access the hypervisor host has to these virtual machines is through the hypervisor's virtual console. This is to allow for better isolation and network segmentation between the hypervisor host and the virtual machines. This also helps to ensure that any potentially harmful experiments don't escape the virtual network, and result in problems on the physical host, or the networks and systems your physical host can access.



4-4: This diagram represents a user accessing a virtual machine on an "internal-only" network. Internal-only networks are more or less host-only networks, but without a virtual network interface installed on the hypervisor host to allow direct network access to the VMs on that network segment. This allows for stronger network segmentation, but restricts users to accessing the VMs on this network segment through their virtual consoles.

4.2 Bare-metal Hypervisors and Virtual Switches

Most bare-metal hypervisors use virtual switches to define networks for VMs to connect to. Bare-metal hypervisors don't exactly have the same concept of virtual network segments that provide different levels of connectivity, like hosted hypervisors. Bare-metal hypervisors typically only have virtual switches with or without an uplink to the hypervisor's physical network card. This means that either the virtual machines connected to a virtual switch are isolated (sort of like an "internal-only" network), or they are seen as systems on the same physical network as the bare-metal hypervisor (sort of like a bridged network segment).



4-5: This diagram depicts two virtual switches configured on an ESXi bare-metal hypervisor. Vswitch1 has no connection to any of the ESXi servers's network interfaces. This means that the virtual machines configured to use this vswitch are isolated, sort of like an "internal-only" network on a hosted hypervisor – the only way a user can interact with these virtual machines is through the virtual console for those VMs available on the ESXi server's management interface (<https://10.0.0.5>). Vswitch2 is uplinked to a network interface connected to the 10.0.0.0/24 network. This allows the virtual machines connected to vswitch2 to interact directly with systems and network services on that network, like how "bridged" network segments work on hosted hypervisors.

Chapter 5 Patch Notes

- Mentioned this in the chapter 4 notes, but I made chapters 4 and 5 change places because it makes more sense to wrap up introductory virtualization concepts in one go, THEN talk about hardware.
- Rewrote much of the chapter, taking more time to go further in-depth about various hardware aspects and how they impact hypervisor, virtual machine, and overall system performance.
- Wrote a blurb about 64-bit support, and how everything /should/ have it, but wanted to be explicit about it.

Chapter 5: Hardware

What kind of hardware is required to run a virtual machine lab? What kind of hardware do you have available? Maybe a used server? Or an old desktop? This chapter is to help readers understand how various hardware aspects will impact your hypervisor choice and performance.

5.1 RAM

All operating systems, software libraries, applications, utilities, your hypervisor (and any virtual machines running) require some amount of system memory (also known as RAM – *Random Access Memory*) to perform their tasks. Like with most computer hardware resources, the more you have, the better your system performs and the more tasks your system can handle at once. Some hypervisors have gimmicky features that allow the hypervisor to dynamically allocate more RAM to systems when it is determined they need it, but I don't recommend relying on them, because they could lead to overextending your available memory, and that can lead to swapping or paging to disk, which will drastically impact system performance. These features are no substitute for having a healthy amount of RAM available to dedicate to each of your virtual machines. The only solutions to a lack of system memory are to either decrease memory load by trying to limit how many things you are trying to do at once, or increase the amount of RAM installed on your physical system and/or the amount of RAM allocated to your virtual machines.

What is Swapping/Paging to Disk?

Most operating systems have a function that allows you to treat a portion of space on a disk drive as though it were a chunk of RAM. This functionality is known by a couple of different names such as a "page file", "swap space" or "virtual memory". If the amount of memory a system requires exceeds the amount of available RAM, it will attempt to utilize that portion of disk space. This process is known as "swapping" or "paging to disk", and causes system performance problems, because it takes considerably more time for the operating system to access the data stored on that "swap space" or "page file" than it does for it to access data stored in RAM.

5.2 Disk I/O

Way back when I was learning how computer components work, there was a very popular demonstration that helped to explain how and why there are so many places computers store data. The image below describes how a computer is like an office:



5-1: This diagram serves a visualization to describe how computers store data, and how disk I/O works. CPU cache is the equivalent of having documents directly in front of you. RAM could be described as documents and materials within arm's reach on the desk, and hard drives are similar to items in filing cabinets or desk drawers – the information is there, it just takes a little bit more time to get it.

CPU cache is data that is directly in front an officer worker, that they are working on right now. RAM would be considered files, documents, and other data the office worker knows they'll need, so it is kept nearby on the desk for easier access. Disk drives are like desk drawers and/or filing cabinets. There's stuff in there that will be needed at some point, but it's stored in the filing cabinet and desk drawers so that the desk isn't cluttered and unmanageable. Not to mention, the filing cabinets can be locked, or may even protect that data in the event of a disaster.

Disk I/O (Input/Output) is described as how long it takes for data to be transferred between system RAM, and the system disk. Think of it as the amount of time it takes to locate stuff in the filing cabinet or desk drawers, and put it on the desk for immediate use, or to remove something from the desk that is no longer needed, and put it into the filing cabinet or desk drawer. Disk I/O comes into play when you are performing a number of tasks that require reading data from disk into RAM or writing data from RAM to disk.

As one might imagine, running a virtual machine is pretty I/O intensive, since the hypervisor has to emulate computer hardware and manage access to chunks of allocated RAM and disk constantly. The I/O requirements only increase as more virtual machines are added to the hypervisor's workload. This means that system I/O can easily become a performance bottleneck, if there are a lot of VMs, system services, or applications trying to access the disk at once. The type of disk drive and number of them installed all have some impact on Disk I/O performance. Let's talk about hard disk drives, SSDs and RAID arrays.

5.2.1 Hard Disk Drives

Hard disk drives are also known as spinning platter drives. They store data using magnetism, and need to spin for the components on the hard drive (read/write heads) to access stored data, or to store data on the disk. All of these things take time – moving the read/write heads over the correct portion of the drive, waiting for the disk to spin up, or spin over the area where the requested data is stored, or is allocated to be written to, etc. The amount of time it takes for a magnetic spinning disk drive to perform these tasks is collectively referred to as *seek time*. This seek time is in addition to the amount of time it takes for the system to communicate across various system buses and controllers in order to store or request data from the drive in the first place.

Most modern hard drives attempt to alleviate this delay through a number of different methods, such as having a small amount of cache memory on the hard drive's controller to store frequently accessed data for faster access, attempting to predict what data the that system will be requesting and pre-emptively retrieving it, or making the magnetic platters spin faster. Most consumer-grade hard drives rotate at 7,200rpm – *rotations per minute*. Some of the more expensive server-grade disks rotate at 10,000 or even 15,000rpm. The faster they spin, the faster the requested data can be accessed or written to disk.

5.2.2 Solid-state drives

Solid-state drives, or SSDs are a relatively new consumer-grade storage (they've been around for a little while on the enterprise side). SSDs are unique in that they have no moving parts and as a result, do not suffer from seek time delay. Overall, they are much faster, and offer much better disk I/O than traditional hard disk drives. The downside to SSDs is that they are a little bit more expensive than standard hard drives. There used to be some concerns about their reliability, but most of these have alleviated through better design and better controller firmware on most solid-state drives doing better housekeeping to ensure that they last longer.

5.2.3 RAID arrays

RAID stands for Redundant Array of Independent Disks. It is a method of improving disk I/O performance and/or system reliability by using hardware called a RAID controller to make a group of disks work together. This can help improve disk I/O by distributing read and write requests to that group of disks. Depending on the type of RAID configuration being used, this can lead to vastly increased performance across the board, but with a reduction in fault tolerance, an increase of read performance and reliability, at the cost of write performance, or some RAID configurations that try to combine the best of both worlds – spreading read and writes across multiple drives, and using parity data to recover from failures, improving fault tolerance. There are wide varieties of RAID configurations, all with varying hardware and disk requirements. RAID

arrays can be composed of hard drives, or solid-state drives, while the RAID controller itself (responsible for managing the RAID configuration and drives) can be built into the system motherboard, or provided through a dedicated hardware controller that the disks are then connected to.

5.3 CPU Cores and Features

All the services and applications running on a system have calculations that need to be done to achieve some sort of input/output, or perform some task. All of those applications and services need a slice of time on the CPU to perform their functions. Generally speaking, the more CPUs and CPU cores you have, the more tasks the CPU can complete at once. Whether that is distributing multiple tasks for a single application across multiple cores or running multiple applications across multiple cores, the more you CPUs and/or CPU cores a system has, the better.

The more resource intensive the applications and services are, the more CPU time they need to perform their tasks. This in turn means that other applications need to wait their turn to run their calculations as well. If you have enough services and system tasks waiting for CPU processing time, this becomes a performance bottleneck, and can result in the system, your applications, and/or your virtual machines becoming unresponsive.

Not unlike with RAM utilization, the only solutions there are to reduce CPU overutilization are to reduce the number of applications or services competing for attention from the CPU, or to increase the number of available CPU cores.

Each virtual machine has at least 1 virtual CPU allocated to it, with there being options to allocate more, depending on the physical number of CPUs and/or cores available on a physical system, and/or limitations put in place by the hypervisor software itself. These virtual CPUs represent a share of your CPU's processing power being dedicated to that virtual machine so that it can run its services and applications.

5.4 Virtualization Extensions (AMD-V, Intel VT-x)

Virtualization extensions are special features only available on select Intel and AMD CPUs. Intel calls their virtualization extensions VT-x, while AMD calls theirs AMD-V. In addition to the CPU, the system motherboard must support and have virtualization extensions enabled as well.

If students are using pre-built systems from a large PC manufacturer (e.g., Dell, HP, etc.), then it's usually as easy as looking up the model name of the computer on the manufacturer's website, and looking for a system manual or a spec sheet to confirm whether or not the system supports virtualization extensions. However, if using a custom-built PC, readers may need to visit the

motherboard and CPU manufacturer's website and search for documentation to verify that both the motherboard and CPU support virtualization extensions.

Intel® Virtualization Technology (VT-x) ‡ ? Yes

5-2: Both Intel and AMD host websites with a complete list of features available for their currently supported CPUs.



5-3: Motherboard manufacturers also usually host user manuals for all of the various models they currently support. The support page for these motherboards usually include links to documentation that tell you if the motherboard BIOS supports virtualization extensions, and how to enable them.

64-bit support (Yes, we're having this conversation)

Most multi-core processors produced by Intel or AMD, most motherboards, and most modern operating systems provide and default to 64-bit operation. Let this serve as a quick reminder to ensure that your CPU, motherboard, and operating system (If using a hosted hypervisor, or attempting to install Client Hyper-V on Windows) are all 64-bit. From here on out, students should assume that their hardware and chosen operating systems will all need 64-bit support.

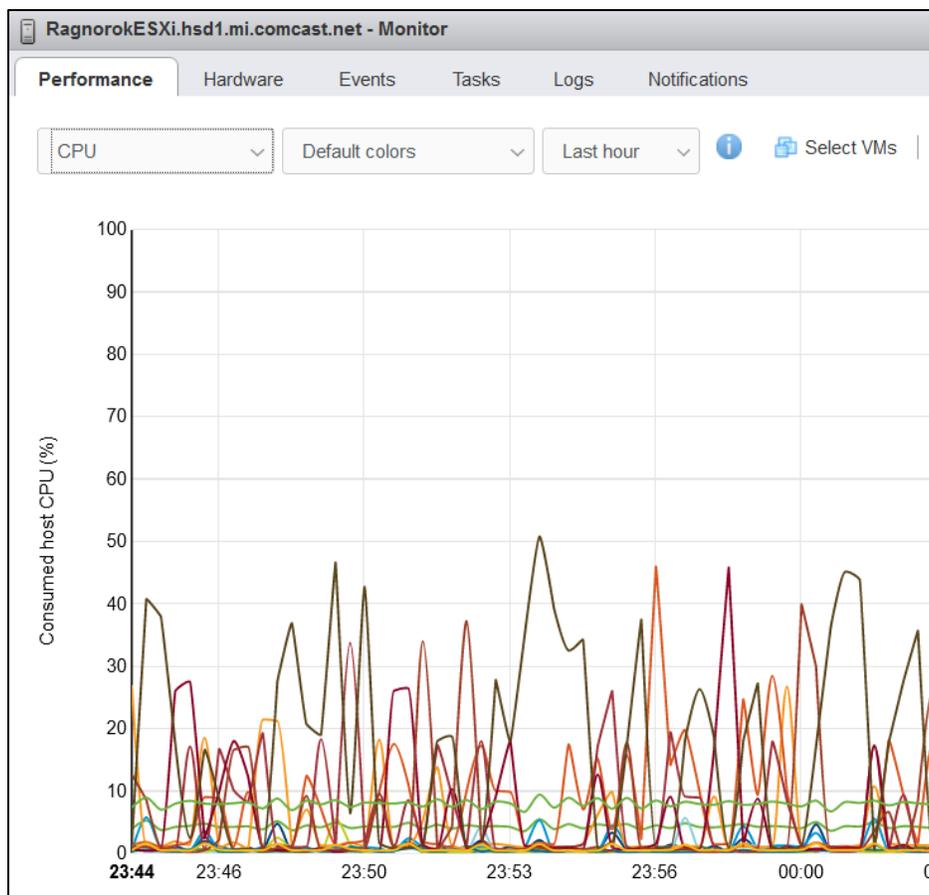
Other Required CPU/Motherboard Features

Most hypervisors have additional prerequisite features, such as Second Level Address Translation (also known as "*Nested Paging*"), and/or NX-bit/Data Execution Prevention (DEP). For the most part, if your CPU and motherboard support Intel or AMD virtualization extensions, it's a safe bet that your system supports Nested Paging. As far as DEP is concerned, it's a feature that's about as old as 64-bit support, so if your CPU has 64-bit support, there's a very good chance that it has NX-bit/DEP support as well. Be sure to check your CPU spec sheet, system documentation and/or motherboard documentation to make absolutely sure your system supports these additional features, if necessary.

5.5 Performance as a Vicious Feedback Loop

This chapter has covered how RAM, Disk I/O and CPU are factors that can impact system stability and performance. A lack of any one of these resources can easily result in a feedback loop that can overburden the others. For example, if there is a lack of RAM on a system, that leads to swapping. Swapping increases demand for disk I/O. This in turn leads to the CPU waiting to retrieve data for the current tasks it is performing, in turn depriving time on the CPU to different service and applications.

Pay attention to CPU, Disk and RAM performance metrics, and either adjust your system load, or increase the available resources according to your needs. Most operating systems have tools available for monitoring performance metrics on your system. Windows has task manager (taskmgr.exe), and the sysinternals suite, while most Unix-like operating systems have a ton of performance measuring utilities that can be ran from the command-line (e.g., top, htop, iotop, free, iostat, sar, etc.). Additionally, most bare-metal hypervisors have their own tools and graphs to help visualize performance of the hypervisor as a whole, or individual virtual machines.



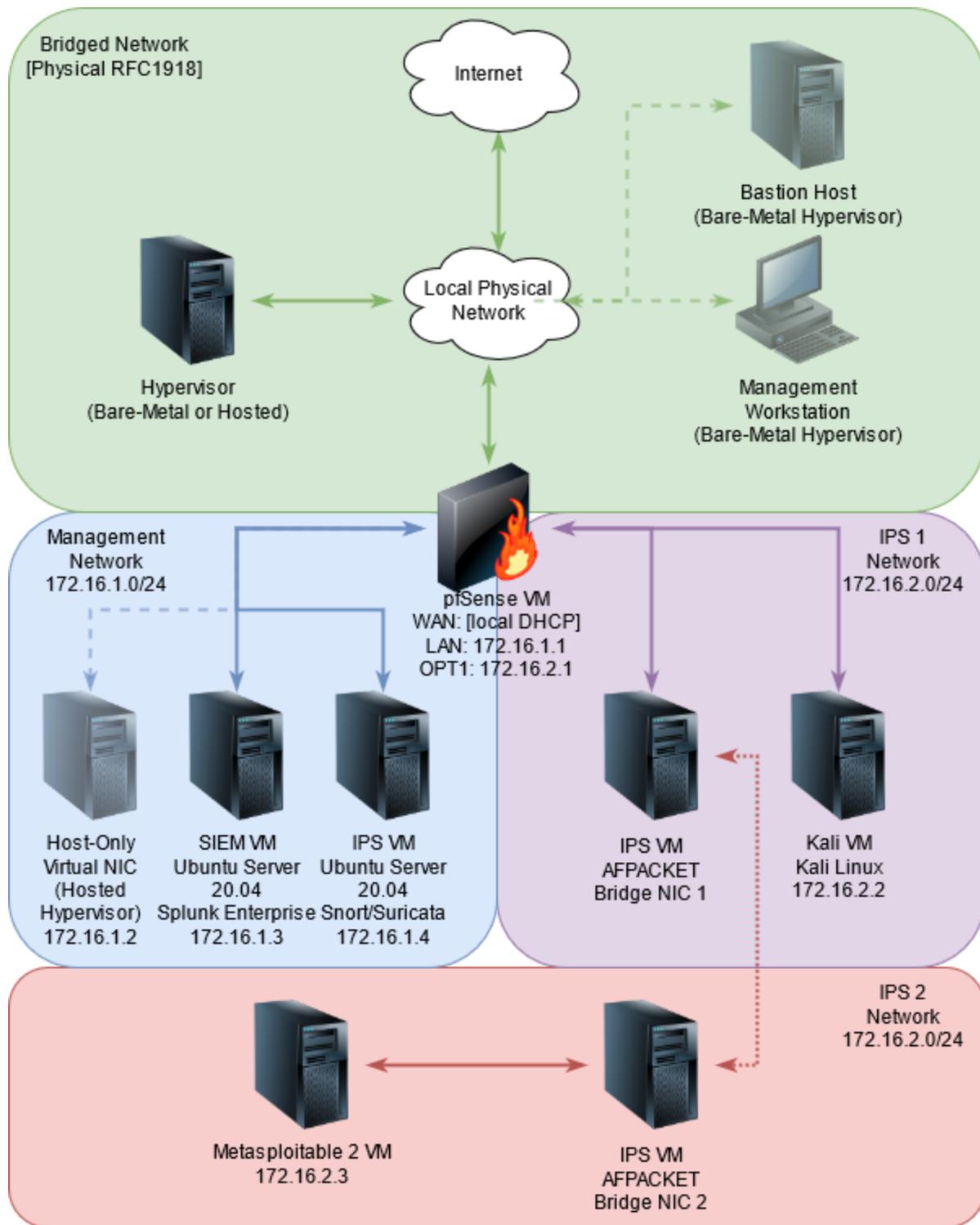
5-4: This is a screen capture from the monitor page on VMware ESXi. The drop-down allows users to select CPU, Memory, Disk or Network to view various performance statistics on the hypervisor.

Chapter 6 Patch Notes

- Re-designed the network diagram. The new diagram includes IP addresses for all of the VMs and clarifies a few of the design aspects.
- Decided to combine chapters 6 and 7 from the first edition.
- Went really deep into the weeds to explain to the best of my ability the purpose of every network segment and virtual machine that is a part of the lab design.
- Included sections on purchasing computer hardware – both what to look for, and where to actually find it

Chapter 6: Virtual Lab Design and Overview

This chapter will focus on analyzing a network diagram of the proposed baseline lab environment students will be building. The goal of this chapter is help students understand the virtual network design, functionality each of the virtual machines will be providing, and the minimum resource requirements to build the lab environment. This lab environment is designed to where it can be fundamentally reproduced on any one of the five hypervisors covered in later chapters. Let's take a look at what the finished product should be:



6-1: The end-product. This is the baseline virtual machine lab that students will produce.

6.1 Lab Network Description – Virtual Machines

The lab consists of 5 virtual machines. Let's start by describing the various virtual machines of our lab environment, and the purposes they will serve:

6.1.1 pfSense

As mentioned in chapter 1, pfSense is a firewall distribution built on top of FreeBSD, and its pf firewall software. Thanks to its intuitive web-based interface known as the WebConfigurator, it is very easy to use, and extremely customizable. The pfSense virtual machine is the keystone of the entire lab environment, sitting between three of the four network segments defined in our network diagram. Its WAN interface will be connected to the Bridged network, the LAN interface to the Management network, and the OPT1 interface to the IPS 1 network. The pfSense VM will be responsible for routing between our various network segments, network access control through the use of firewall rules, and providing DHCP, DNS, NTP, and HTTP proxy network services for our lab virtual machines.

6.1.2 SIEM

SIEM is an acronym that stands for Security Information and Event Management. This system is going to allow students to more easily collect and analyze logs from the IPS virtual machine. Our SIEM VM is going to be based on the Ubuntu Server 20.04 Linux distribution, and will have Splunk log management software installed.

Splunk is described as "software for searching, monitoring and analyzing machine-generated big data, via a Web-style interface." Buzzwords aside, that means that Splunk is software that is used to collect, index, and analyze logs generated by a wide variety of sources. There are three core components to a Splunk installation: A web front-end (called a searchhead) that users can use to query collected logs, an indexer that serves as a collection point for logs gathered from your environment, and log forwarding software installed and configured on systems you wish to collect logs from. The SIEM VM is going to serve the role of searchhead, and indexer. We will be forwarding logs from the IPS VM.

Splunk is not considered free software, but the company provides a trial edition that will be utilized for the lab environment. The trial edition is only limited by how much data users are allowed to index per day – 500MB. This *should* be enough indexing per day to serve the needs of the lab environment.

6.1.3 IPS

IPS is an acronym that stands for Intrusion Prevention Software. Other related acronyms (that are more or less interchangeable) include Intrusion Detection Software (IDS), Network Intrusion Detection Software (NIDS), and Network Intrusion Prevention Software (NIPS). IDS software works by analyzing network traffic and matching it against special patterns or signatures called rules. If network traffic the IDS has observed matches one of if the rules it has defined, then usually an alert is logged with details about the rule that was triggered, and the connection in which the pattern was observed – source and destination IP addresses, transport protocol, source and destination ports, etc. Sometimes, these alerts will also occasionally include a packet capture of the traffic for further analysis as well.

The IPS VM is going to be based on Ubuntu Server 20.04. Readers will be given the choice of installing one of two open-source IDS software projects – Snort3 or Suricata. Students will then be provided with a script that will handle installing and configuring their chosen IDS software for use in their lab environment. In addition to their choice of IDS software, students will be instructed on how to install a Splunk universal forwarder and configure it to forward Snort3, or Suricata IDS alerts to the SIEM VM.

6.1.3.1 AFPACKET bridging, and Fail-Closed Networking

In [figure 6-1](#) (p. 58), you may have noticed that there are three entries for the IPS VM. One entry on the *Management Network*, another entry on the *IPS 1 Network*, and a third entry on the *IPS 2 Network*. These entries are all the same virtual machine with virtual network cards connected to all three of those network segments. The network interface on the Management Network is the interface students will be able to interact with, if they choose to enable remote administration through the SSH protocol, and will be the interface used to forward IDS logs to the SIEM VM. The other two interfaces will be used exclusively by Snort or Suricata in AFPACKET bridging mode. These interfaces will not receive an IP address and will not be accessible for remote administration.

So long as the IPS virtual machine, and either the Snort or Suricata software is operating correctly, the IPS 1 and IPS 2 network segments will be bridged together. The two network interfaces on those network segments will operate like a layer 2 network bridge, transparently forward traffic to and from the IPS 2 network segment. The primary difference between bridged virtual network segments (described in chapter 4, [section 4.1.1](#), p. 45) and the AFPACKET bridge the IPS VM will be providing, is that this bridging functionality can be enabled and disabled at will, very quickly. We can use the IPS VM to implement something called fail-closed networking.

So long as the IPS VM and the IDS software is running, the IPS 1 and IPS 2 networks are bridged, right? That means if we disable the IDS services, or immediately power down the virtual machine, that bridge no longer exists. The network connectivity fails, and hosts on the IPS 2 network are isolated immediately. These can be extremely useful for temporarily enabling network access, or

instantly isolating a selection of virtual machines connected to the IPS 2 network segment. If it helps, think of fail-closed networking as a sort of dead man's switch. The connectivity is only available so long as the Virtual Machine and IDS software are both operating. Let's explore some examples where this could be helpful for limiting your risks.

A student has opted to convert their IPS 2 network into a penetration testing lab. The AFPACKET bridge on the IPS VM can be temporarily enabled to allow virtual machines with penetration testing tools to update, then disabled when those updates are complete to isolate the network segment again. This limits the risk of penetration tools causing problems for other network devices outside of the lab environment.

As another example, a student has opted to convert their lab into a malware analysis network. They've installed a virtual machine with malware analysis and sandboxing software on the IPS 2 network. In this instance, having external network access is important to observe what resources or network addresses an unknown malware sample will attempting to connect to. However, something has gone wrong, and the malware is attempting to attack other systems it has identified over the network. The IPS VM can be shut down to immediately isolate the IPS 2 network and stop the malware from attempting to spread.

6.1.4 Kali

Kali is a Linux Distribution packed with network penetration testing software. It is considered the standard for network penetration testers and skids around the world. Not because it is necessarily better than other network penetration testing software distributions, but because its relatively well-supported due to its sheer volume of users. Having so many users means that even if the official support channels don't have the answers you need, or are unhelpful, the large user community is likely to have experienced, or perhaps even documented and resolved some of the same problems students may be facing. You know, monkeys and typewriters.

The Kali Linux VM exists to perform two tasks for our lab environment. The first task is to confirm whether the AFPACKET bridge between the IPS 1 and IPS 2 network segments that the IPS VM is providing works as intended. Students will be testing the bridge by performing tasks to enable or disable the bridge, launching network probes towards the metasploitable 2 VM on the IPS 2 network, then observing the results. The second task is to verify that the IDS software on the IPS virtual machine is capable of both logging network attacks, and forwarding those logs to the SIEM VM. This will be accomplished by using the Kali's penetration tools to simulate an attack against the Metasploitable 2 virtual machine, and observing those results.

6.1.5 Metasploitable 2

Metasploitable 2 is best described as a network security punching bag or a "shell piñata". It is very old, very vulnerable to attack, and serves as a rite of passage for most network security practitioners. At some point, most newcomers to security operations and/or network penetration testing are told to try out "boot2root" exercises. These are intentionally vulnerable virtual machines that are used to impart different security concepts through hands-on interaction (and sometimes, a lot of reading of the exploitation guides if you get stuck). Metasploitable 2 is probably one of the oldest and most well-known boot2root virtual machines out there. Like the Kali VM, Metasploitable 2 exists to perform two key tasks for our lab environment: Verify the AFPACKET bridge provided by the IPS VM works as intended, and verify whether or not network attacks launched by the kali VM are being logged and forwarded to the SIEM VM.



6-2: The best description of Metasploitable 2. Yes, I know that's the Windows Logo. Meme format courtesy Ben Heise. Rick and Morty property of Cartoon Network, Inc. Support the official release, etc.

6.2 Lab Network Description – Network Segments

In addition to the virtual machines described above, the lab is divided into 4 network segments, and three unique subnets.

6.2.1 Bridged (Physical) Network

The Bridged network segment is going to correspond to the local network students utilize at home, work, or school. This is the network where your hypervisor (be that a bare-metal or hosted hypervisor) is going to be located. For readers who will be creating their lab environment on VMware ESXi (bare-metal hypervisor), there are objects in the bridged network segment to represent the management workstation, as well as a dedicated bastion host, if necessary, to maintain access to the bare-metal hypervisor and the virtual machines hosted on it. Bastion hosts are covered in-depth in Chapter 16, *Routing and Remote Access for Bare-Metal Hypervisors* (Beginning on p. 835).

6.2.2 Management Network

The management network corresponds to 172.16.1.0/24. As the name implies, this is a network that is dedicated to managing most of the virtual machines of the lab environment. Hosted hypervisor users will have a direct connection to this network segment through the use of a virtual network card attached to the hypervisor host (assigned the IP address, 172.16.1.2 in the network diagram). This enables easy access to the most of the virtual machines in the lab environment. In addition to easy access, this makes the lab environment on hosted hypervisors extremely portable, meaning that the hypervisor host can be moved from one physical network to another, with minimal impact to network access or functionality of the virtual lab environment.

6.2.3 IPS 1 and IPS 2 Networks

IPS 1 and IPS 2 share the network, 172.16.2.0/24. As mentioned in section 6.1.3, these two networks are connected through the AFPACKET bridge supplied by the IPS VM. This functionality is what enables these two networks to share one network range.

6.3 Resource Allocations, and Hardware Requirements

Here is a list of recommend CPU, Memory and Disk Space allocations for each virtual machine:

- PfSense: 512MB RAM, 5GB Disk, 1 CPU/core
- SIEM: 4GB RAM, 80GB Disk, 1 CPU/core
- IPS: 4GB RAM, 80GB Disk, 1 CPU/core
- Kali: 4GB RAM, 80GB Disk, 1 CPU/core
- Metasploitable 2: 512MB RAM, 10GB Disk, 1 CPU/core

These resource allocations ensure that the virtual machines perform adequately for a personal lab environment. Adding the disk and RAM allocations for all 5 virtual machines comes to 255GB of disk space, and 13GB of RAM. Note that this does not include additional disk space for snapshots, the hypervisor itself, storing installation ISOs, or disk space and RAM for the host operating system (if using a hosted hypervisor).

My recommendations are to use a system with at least 500GB of free drive space, 16GB of RAM, and at least 4 physical CPU cores (Hyperthreading doesn't count). This also assumes your system supports all of the CPU and motherboard features necessary to run most modern hypervisors (64-bit operation, NX/XD bit support, and either AMD-V or VT-x).

If the recommended system specifications are too steep, experiment with reducing the amount of RAM and disk space allocated to the SIEM, IPS and/or Kali virtual machines, starting with the Kali VM first, because it is not considered critical to the core function of the lab environment. If at all possible avoid allocating less than 2GB of RAM and 60GB of disk space to the Kali, IPS, or SIEM VMs.

I'm in the market for new hardware. What should I buy?

Another alternative to adjusting resource allocations on your virtual machines would be to purchase dedicated hardware for your lab environment – if your budget allows for that. Before you begin, consider the goal of your purchases, and the budget you have versus the hardware you want.

Are you planning on purchasing desktop PC parts to put together a server to host your VMs? Remember that bare-metal hypervisors can be extremely picky about the hardware they support.

Are you on the lookout for a used server because overkill is the best kill? Make sure to take power, cooling, and the insane amounts of noise rack-mount servers make into your considerations.

Are you looking for a system to host your VMs that is both powerful and portable? Save every penny you can for either a high-performance laptop, a shuttle form-factor PC, or an Intel NUC (or equivalent book-sized computer), because small form-factor systems are expensive, but small form-factor systems that are actually worth your hard-earned money are even more expensive.



6-3: Like with most purchases, have a budget prepared, an end-goal in mind, and be sure to weigh the pros and cons before committing. For example, used rack-mount servers are super powerful, but the electric bill, cooling requirements and sheer amount of noise generated by small fans spinning at a very high RPM can put out can be very off-putting.

Where can I find cheap hardware?

Over the course of many years and advice from many friends and acquaintances, I've been provided a list of places to purchase hardware on the cheap. I've done my best to hopefully provide useful recommendations on where you might find good hardware at a bargain. Here is a list of well-known websites, and electronics retailers to consider:

- Newegg.com
- Slickdeals.net
- SaveMyServer.com
- ServerMonkey.com
- Natex.us
- TheServerStore.com
- Microcenter

Unfortunately, some of these vendors only serve the United States, or North America. But all hope is not lost, there are some local options to consider:

E-bay/craigslist/other online marketplaces: I was on the fence about including these as potential sites or places to acquire cheap hardware, for a number of reasons -- Some sellers can be extremely sketchy, scams can sometimes run rampant, quality of the product can vary greatly, etc. I still decided to include them as recommendations anyway, because you have the option of buying locally, seeing what it is you are purchasing, and testing it before you put up hard-earned money for your hardware. I Also wanted to take a moment to mention reddit.com/r/homelab as a potential resource for both homelab guidance, as well as a potential source for acquiring hardware.

Government surplus and/or government auction sites: government surplus and/or used hardware can end up in all sorts of the places. You might get lucky and your regional or national government might even have a surplus/auction website where you can bid on hardware. For example, in North America, there's GovDeals.com where you can search for computer hardware narrowed by region to see if there is anything worth snagging in your area.

Electronics/E-Waste recyclers: Sometimes, when computers have reached the end of an equipment lease, data sanitization and resale is contracted out to local electronic recycling facilities and can lead to acquiring hardware at a nice discount.

Local Colleges/University surplus: You may be able to contact a local college or university and talk with people responsible for asset management and get pretty good used hardware. Some universities even have websites where you can purchase their surplus hardware.

Chapter 7 Patch Notes

-New chapter. I felt like if I'm telling users to download a password manager, I should probably explain why and how to use it.

Chapter 7: The Importance of a Password Manager

In chapter 1, students were provided with a huge list of applications to download. One of those items was a password manager called *KeePassXC*. A password manager is an application used for storing credentials. If it's a service or system that requires a username and/or password, a password manager can generate, store, and remember that information for you. The password gets stored to an encrypted database, and in order to access that database, a master password is required. Password managers sound really awesome, but have pros and cons that need to be considered before going all-in. This section will be dedicated to teaching students about the pros and cons of a password manager, guide you through the steps of creating a KeePassXC database, and creating their first password database entry.

7.1 Benefits of Password Managers

One of the many pros of using a password manager is that it never forgets a password. Store the credentials in the password manager, save the database file, and that's it. When a user wants to log in, they open their password manager with their master password, select the correct entry in the database, and copy the credentials.

Another perk is that most password managers have a feature called a password generator. Most places have rules for creating a password called password complexity requirements. They're meant to make your passwords harder to guess in order to prevent unauthorized access. Most of the time, people find them annoying and will pick the simplest password they can remember that meets the requirements, so they can go about their day. For example, [Capitalized Current Season][Current Year][Special Character(s)] is a VERY common pattern. It leads to passwords that are easy to guess like "Winter2016!?". That password is 12 characters in length, has a Capital letter, multiple numbers, and two unique special characters. It *has* to be good, right? Well, the system accepted that password, so it's good enough! Except, it really isn't. Password generators allow users to avoid this problem by selecting a couple of password complexity settings, and letting the password manager randomly generate and store a suitably long and complex password on their behalf.

By generating strong passwords for every site and service they log in to, password managers and generators help users to avoid password re-use. As the name implies, it is the re-use of credentials across systems or services that a user has access to. Password re-use presents an opportunity to where an adversary only has to correctly guess one username and password. They can then attempt to re-use that set of credentials for other services the user has access to. This type of attack is referred to as credential stuffing. Attackers will take a set of usernames and

passwords exposed from a data breach, or account compromise, and attempt to re-use those credentials to gain unauthorized access to other systems and services.

7.2 Weaknesses of Password Managers

A major weakness of using password managers is that they can become a single point of failure or risk, if certain problems are not mitigated. One of those weaknesses is the master password itself. The master password is used for accessing the entire database of stored credentials. What happens if the master password is weak, and an adversary gets a copy the password database? All of those neat features and positive aspects of a password manager go flying out the window, because the most important password was also the easiest one to guess.

Another risk factor is storage of the password database file itself. Where is the password database stored? Is that storage secure? Are there multiple backups? These are all important questions to ask, because if the system or storage media used for storing the password database is lost, stolen, destroyed, or otherwise damaged, the database, and all of its credentials, are lost with it. If the password database is stored in the cloud (either manually by the user, or through the password manager itself automatically), *who really has access to that password database?* The answer is, *I have no idea*. The cloud is somebody else's computer. Companies will show you all sorts of paper trails and audits that say the confidentiality of your data is guaranteed, but you have no way of validating that.

One last risk to consider before we move on is that of cross-contamination. Password managers are very convenient. Sometimes, that leads to users storing passwords for their private accounts in the same database where they store credentials for accessing work-related systems and services. Cross-contamination is a two-way street. If an adversary gets access to a password database that has both work and home credentials, they could use those credentials to attack that person's workplace, or potentially commit fraud and identity theft if the password database includes access to bank accounts, credit cards, or services that store that information, or other PII (Personally Identifiable Information).

7.3 Mitigating the Weaknesses

The good news is that awareness of these problems can lead to ways of mitigating them. Users can set a strong master password (or use multiple forms of authentication) to protect the password database. Most password managers will allow you to back up your password database files manually. That way, users have control of the data, and know where another copy can be found in an emergency. Multiple password databases can be created to prevent cross-contamination. While having to remember more than one master password is annoying, it's easier to remember two or three slightly more complex passwords, instead of hundreds of individual passwords for various systems and services.

Why are We Having This Discussion?

Now, some readers might be thinking: *It's just my lab environment. Why are we talking about this? What does any of this have to do with VM labs?* Well, the answer is because in spite of the risks and drawbacks associated with password managers, the benefits vastly outweigh them. In fact, some environments require (or heavily encourage) their employees to use them – and for good reason. Most of us are required to manage a massive number of credentials in both our personal and professional lives. And while there are other forms of authentication out there, passwords are universal – and they are inevitable. They've been around for the entirety of human history, and chances are they'll still be here long after we're gone, no matter how many "password killers" there are. So, **managing your passwords safely is a good habit worth learning.**

I chose *KeePassXC* because the price is right (free), the user keeps control of the password database (doesn't get sync'd to the cloud, unless users do that themselves with cloud storage applications), and it's available across every major desktop operating system out there (Windows, Linux, and MacOS). I'm not going to cover installation instructions for *KeePassXC*, because it's very straightforward – Windows users get the option of a portable version that doesn't require installation or an MSI installer. MacOS users can use the DMG installer, or if you are a power user, you can install it via homebrew. Linux users get the choice between an app-image (which is sort of like the portable version that Windows users get) or retrieving it through their distribution's package manager. I'm assuming that if you use Linux as your desktop operating system you know how to operate your distribution's package manager. For those already familiar with password managers, you are more than welcome to use what you are most comfortable with.

For those who don't want to use a password manager at all, you could consider storing your credentials in a physical notebook. Most security people would consider this bad advice (probably because they've been burned by users storing passwords on sticky notes stuck to their monitors, or on pieces of paper stored under keyboards), but I disagree. A notebook of passwords can be just as effective as a password manager, provided that access to it is limited, and it's stored safely when not in use.

One last thing: The next two sections are about creating a password database and password entries in *KeePassXC*. The instructions will be demonstrated using the Windows client, because I'm a filthy Windows user. However, this should not matter. The interface, icons, and configuration options are consistent regardless of the operating system readers prefer to use, making it easy to follow along.

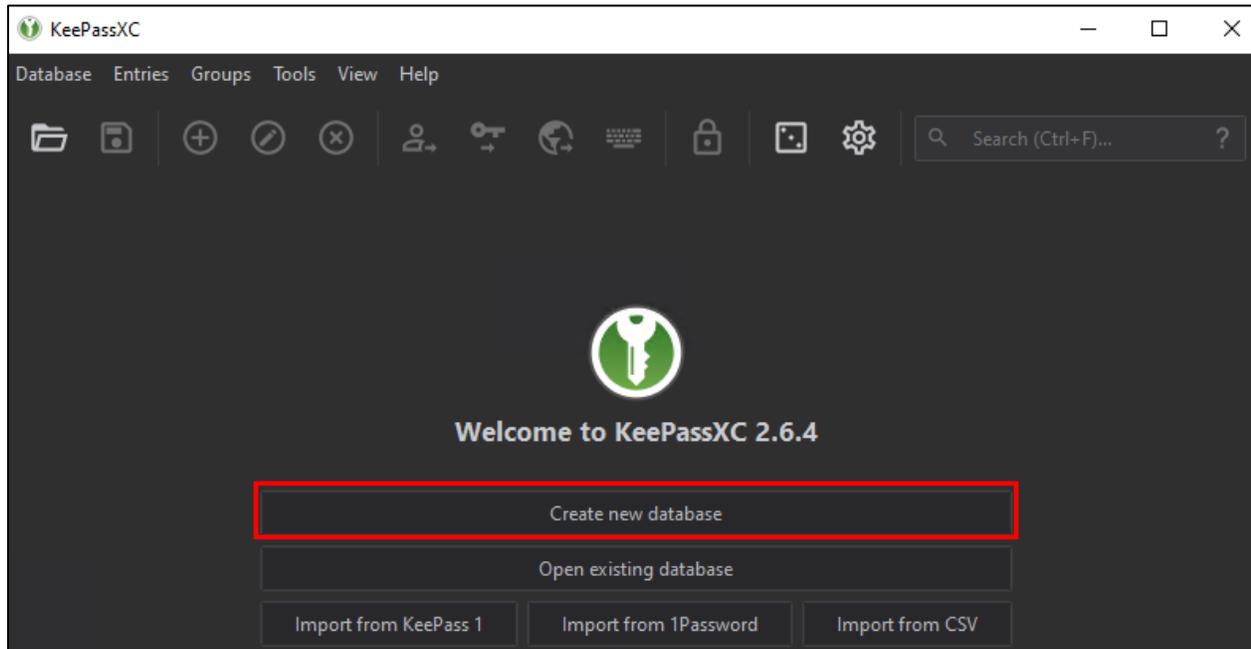
7.4 Creating a Password Database File with KeePassXC

When students open *KeePassXC* for the first time, they'll be greeted with the welcome screen. To create a new database, students can click the *Create new database* button, or select *Database > New Database* in the navigation menu. A window labeled *General Database Information* will appear with two input boxes, labeled *Database Name*, and *Description*. After entering a name for the database and description of its contents, click *Continue*.

On the next screen, labeled *Encryption Settings*, students can customize the strength of the encryption and the format of the database file. The default options should be sufficient, but readers are welcome to explore and customize further on their own. Click *Continue* to proceed.

The next screen, labeled *Database Credentials*, allows users to set the master password for the database. The icon that looks like an eye can be clicked to reveal the contents entered on both input boxes. The *Add additional protection* button adds new options that allow users to set up additional security measures for unlocking the database. Users can require a special key file, or set up a YubiKey along with the master password to provide multi-factor authentication. We will not be exploring the additional protection options. Enter the password in the *Enter password* input box, and again in the *Confirm password* input box, or students may utilize the password generator to allow KeePassXC to generate a password or passphrase automatically. The square icon with three diagonal dots (like a die) opens the password generator dialogue to generate a random master password for the database. Check out the sidebar conversation below, *Password Generator Crash Course*, to learn more about how to use the password generator. When finished, Click *Done* to create the new password database.

After clicking *Done*, users will be prompted through their operating system's file manager for a location to store their password database. Navigate to where you would like to store the database, and click *Save*.



General Database Information

Please fill in the display name and an optional description for your new database:

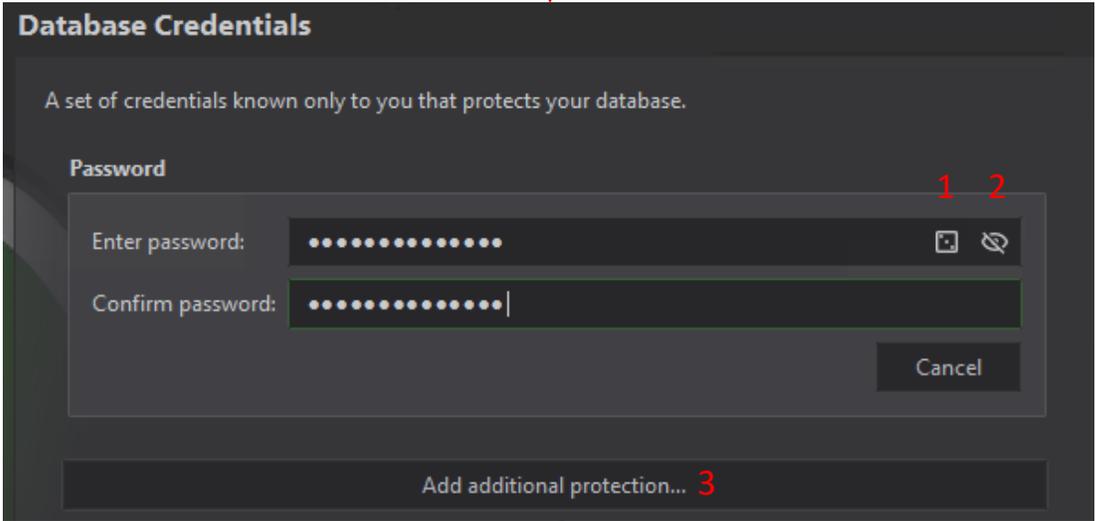
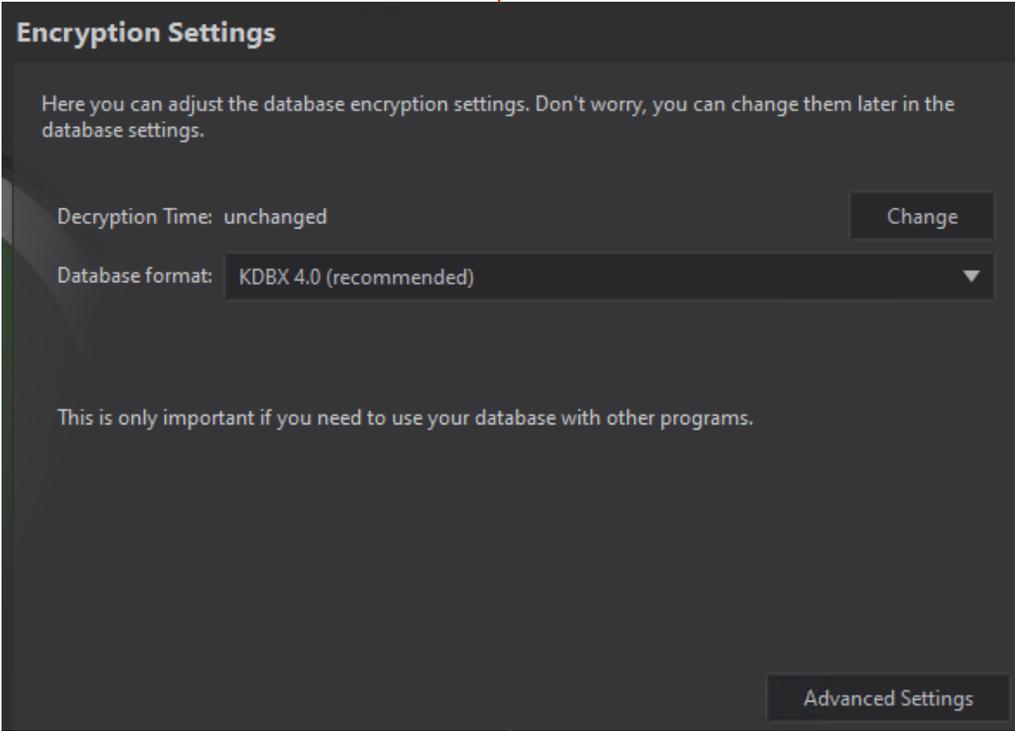
Database Name:

Description:

Continued to *fig. 7-2*

7-1: Open KeePassXC, then click on the Create new database button. This opens a new window, and wizard that guides students through creating their first password database. The first screen of the wizard, labeled *General Database Information* asks students input a name and description for their database.

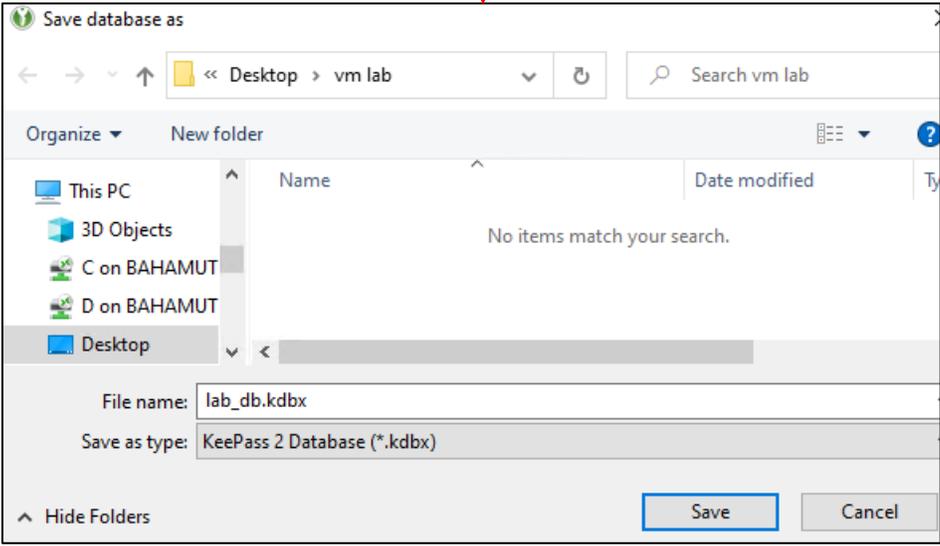
Continued from *fig. 7-1*



Continued to *fig. 7-3*

7-2: On the Encryption Settings screen, students can customize how encrypt is configured for the password database. Accept the defaults by clicking *Continue*. The Database credentials screen is used to assign a password to the password database. Enter a secure password or passphrase into the *Enter* and *Confirm password* input boxes. The icon that looks like a die (1) can be used to open the password generator, while the icon that looks like an eye with a slash through it (2) can be used to reveal the password students entered into the input box. The *Add additional protection* button below (3) can be used to assign a key file and/or Yubikey auth in order to access the password database. When finished, click *Done* to complete the wizard.

Continued from *fig. 7-2*



7-3: Clicking *Done* will prompt students for a location to save their password database. Choose a location for the file, and click *Save*. Get in to the habit of making copies and backups of the password database on regular intervals.

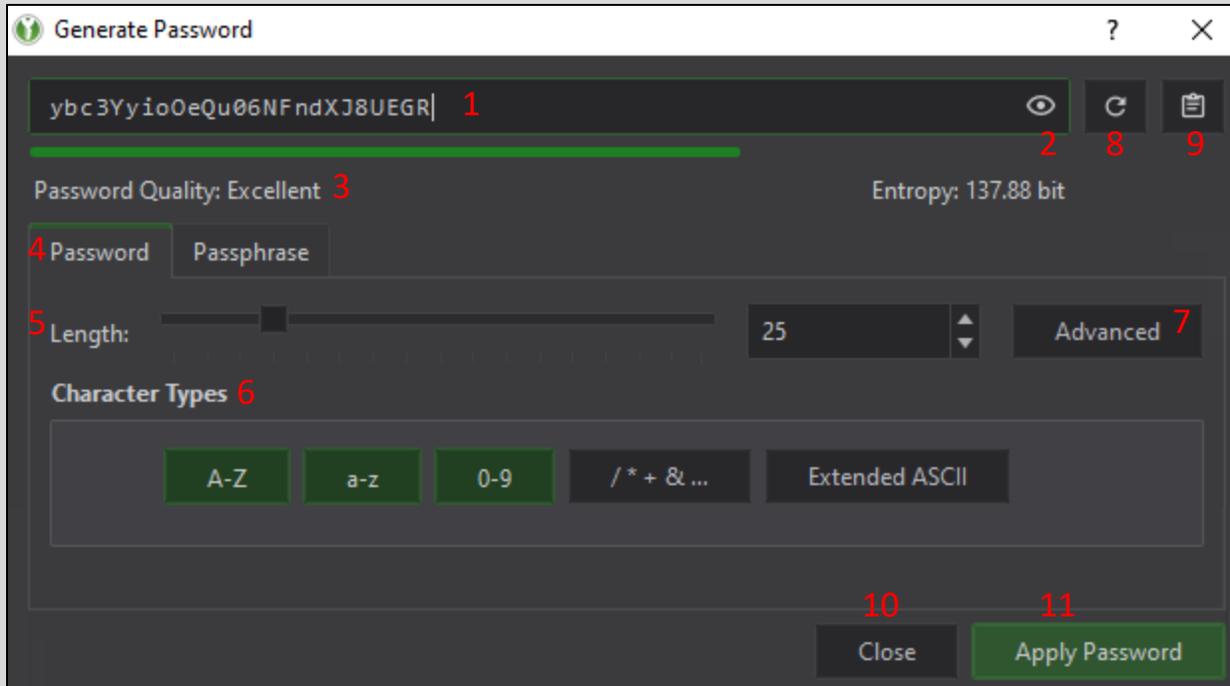
Password Generator Crash Course

The *Generate Password* window is accessed through the die icon seen in figure 7-2. The input box located near the top of the window where generated passwords will be displayed. Since it is an input box, users have the option of adding additional input to customize the password as necessary. For example, if the password generator creates a long password, but doesn't meet one of the complexity requirements for a site or service, you can add characters to the password manually. You will need to click the slashed eye icon in order to reveal the generated password.

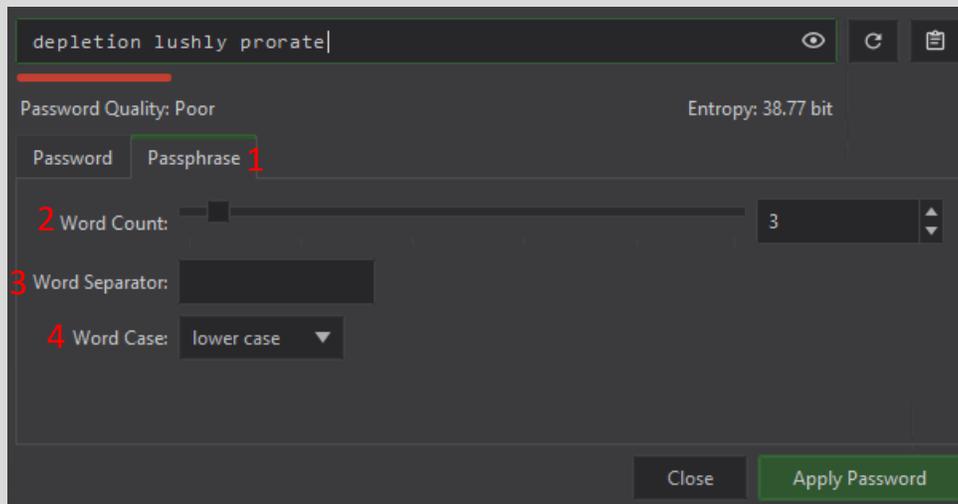
Under the input box and password strength analysis are two tabs labeled *Password* and *Passphrase*. By default, the *Password* tab is selected. This tab allows you to set various options and parameters for generated passwords. You can set the maximum length through the slider bar labeled *Length*, or the input box with the up and down arrows to the right. The section labeled *Character Types* allows you to specify what characters are allowed in the password *KeePassXC* will generate for you, such alphanumeric (upper and/or lowercase letters), special characters, etc. The highlighted buttons are characters sets selected for generating passwords. Clicking the *Advanced* button reveals more parameters to control what characters can be used for password generation.

The *Passphrase* tab allows users to create passwords based on a collection of random dictionary words. The *Word Count* slider bar and input box allows users to control how many words the generated passphrase will include. The *Word Separator* input box defines what characters will be used to separate each word in the passphrase. Lastly, the *Word Case*: drop-down allows users to generate a passphrase with all lowercase, capitalized, or title case words.

To the right of the *Password/Passphrase* input boxes, are two buttons. The circular arrow icon causes the password generator to "reroll" and create a new random password or passphrase. The clipboard icon will copy the password/passphrase to your operating system's clipboard. Finally, near the bottom of the window are the *Close* button which closes the window, and the *Apply Password* button. *Apply Password* will automatically close the password generator window and copy the Password/Passphrase in the Password input box to both the *Enter Password* and *Confirm Password* input box in the previous screen (in our case, the *Database Master Key* screen, but later on, back to the *Add entry* window when adding credentials to your database). Notice the KeePassXC interface provides direct access to the password generator through a dice icon on the main screen. The only difference is that when accessing the Generate Password window through the icon on the main window, there is no *Apply Password* button. Passwords will need to be copied to the clipboard, and pasted elsewhere as necessary.



7-4: Welcome to the Generate Password window, there are tons of option that control password generation options for KeePassXC. First is the password input box (1). Click the eye (2) to reveal the password. Passwords automatically have their quality and strength(entropy) measured (3). Users can choose to generate a Password or a Passphrase (4), with the *Password* tab selected by default. The *Length* slider and input box (5) can be used to define how long the password should be. The options under *Character Types* (6) tells the generator what characters can be used to generate a password. Users can access more character set options through the *Advanced* button (7). Users can *Regenerate password* and "reroll" the generated password/passphrase (8), Copy it to the system's clipboard (9), click the *Close* button (10) to close the window, or click *Apply Password* (11), to close the password generator and jump back to the previous screen, filling in the enter and confirm password input boxes.

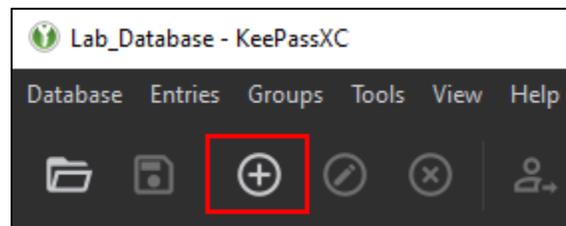


7-5: The Passphrase tab (1). Users can define how many words to use for their passphrase (2), what characters to use to separate each word (3), and what letter case they want for the words of the passphrase (4).

7.5 Creating Password Database Entries with KeePassXC

Once students have created their database, or entered the master password again to access it, you'll be in the database management screen. Click the circle with a + in the middle towards the top of the window, or you may select *Entries > New Entry* from the navigation menu.

The *Add entry* screen appears and has a collection of input boxes students can use to describe the credentials they wish to add to their database. Most of the input fields are pretty self-explanatory, such as *Title*, *Username*, and *URL*. The *Notes* field is a freeform input box you can use to store information about this entry. For example, documenting an API key for a web service so it can be retrieved without needing to log in to the user account on that particular service (for example, snort.org oinkcodes). Or the field could be used for documenting OS and network information for a particular system. Above the *Notes* field is a checkbox labeled *Expires*. If this box is checked, the user can set a date to mark this entry as expired. This function can be used as a reminder if users are required to change passwords regularly. That brings us to the *Password* field, where you enter the password to store for this entry. To the right in the Password input box are the eye icon that can be used to reveal the password, and the die icon that can be used to open the *Generate Password* window. When finished, click the *OK* button to save the new entry. *KeePassXC* automatically saves the password database to disk after adding new entries, or editing existing entries.



Continued to *fig. 7-7*

7-6: With a password database created, users can begin populating it. Click the icon in the square, or select *Entries > New Entry* from the navigation menu to get started.

Continued to *fig. 7-6*

Root · Add entry

1 Title: ADVENT network relay

Username: mor_balaten 5 4

Password: ●●●●●●

URL: https://www.youtube.com/watch?v=DFDSFcyMpFg

2 Expires: 5/5/2021 2:24 PM Presets

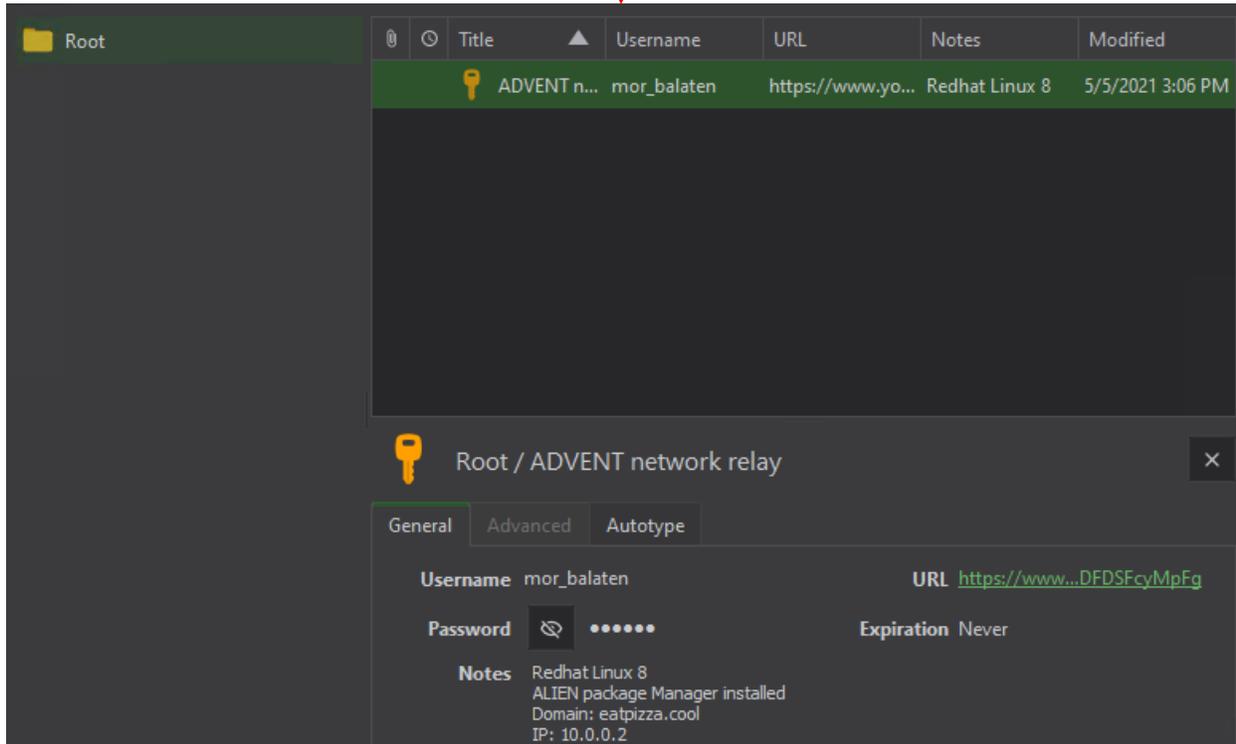
3 Notes: Redhat Linux 8
ALIEN package Manager installed
Domain: eatpizza.cool
IP: 10.0.0.2

OK Cancel

Continued to *fig. 7-8*

7-7: The *Add entry* screen. Most of the input boxes are pretty self-explanatory. It's a good idea to add as much information as you can – including a descriptive title for the entry itself (1), the next expiration date if this credential needs to be renewed regularly (2), and other contextual notes as necessary (3). This screen also includes the icon to reveal the contents of the *Password* field (4), and the die icon to open the password generator for this entry (5). Click *OK* to save the new entry to the password database.

Continued from *fig. 7-7*



7-8: After clicking *OK*, the new entry is added to the Root folder of the database, and *KeePassXC* immediately saves the database to preserve new entries or modified entries.

Chapter 8 Patch Notes:

- Better explanations on the differences between each of the hypervisors
- Explained the difference between Hyper-V Server, Windows Server with the Hyper-V role, and Client Hyper-V
- Explained the difference between the standard and professional editions of VMware Workstation and Fusion, and why we need the professional editions (creation of custom virtual networks)
- Vastly improved general chapter outline for the hypervisor setup chapters. Students should know exactly what to expect when jumping to the chapter for their hypervisor. It also serves as a template in case students want to get creative and use a hypervisor that isn't covered.
- Blurb on content and why it seems so repetitive
- Added a section to make sure users know I'll have guidance and instruction for all desktop operating systems.

Chapter 8: Time to Choose Your Destiny

It's time for students to determine what hypervisor they'll be using to build their lab environment. The next 5 chapters of this book are guides on how to create the lab environment described in chapter 6, on one of five different hypervisors. Three of these hypervisors are considered hosted, while two are considered bare-metal. Pick one of the following hypervisors to get started.

8.1 Hypervisor Choices

Microsoft Client Hyper-V: A bare-metal hypervisor provided by Microsoft. Client Hyper-V is somewhat unique in that it is classified as a bare-metal hypervisor, but behaves like a hosted hypervisor. It is a free feature that can be enabled, but only specific editions of Windows 8.1 (Professional or Enterprise edition) or Windows 10 (Education, Professional, or Enterprise edition). Jump to [chapter 9](#), starting on page 86, to get started.

Hyper-V Server, Hyper-V Windows Server Role and Client Hyper-V: What's the difference?

Client Hyper-V is a feature for select Windows desktop operating systems. As mentioned above, it's a bare-metal hypervisor, but it has the ease of use of a hosted hypervisor, allowing users to manage it either via the command-line (PowerShell) or through the *Hyper-V Manager* desktop application on the same local system, or over the network.

Windows Server also has a version of Hyper-V, installed as a "server role". Once again, it is a bare-metal hypervisor, but behaves much like a hosted hypervisor, easily managed and configured through either the command-line or via the Hyper-V Manager application.

Finally, that brings us to Hyper-V Server. This version of Hyper-V is probably the closest to behaving like a traditional bare-metal hypervisor. It isn't a feature or role that is installed on a desktop or server operating system, but a barebones operating system, containing the hypervisor itself, and a command-line interface for doing initial network setup so that the server can be managed remotely.

This guide is specifically for Client Hyper-V running on Windows 10. Readers of the first edition have reported that the instructions are very easily adapted to the Hyper-V role on Windows Server installations, and hypothetically, readers could adapt the concepts from the chapters 16 (*Routing and Remote Access for Bare-Metal Hypervisors*) and 14 (*pfSense Firewall Rules and Network Services Guide – specifically, Firewall Rule configuration for Bare-metal Hypervisors*) to reproduce this lab environment on Hyper-V Server. My only disclaimer is that these Hyper-V installations are not (yet) officially covered in this book, and your mileage may vary.

Oracle VirtualBox: A hosted hypervisor provided by the Oracle Corporation. VirtualBox is free and available for Windows, and practically any Unix-like operating system (Linux, BSD, MacOS, etc.). Navigate to [chapter 10](#), starting on page 196 to get started.

VMware Fusion Pro: A hosted hypervisor by the VMware Corporation for MacOS. The lab environment specifically requires Pro edition. If you'd like to know why, read the sidebar, *Why Are We Going Pro?* Otherwise, go to [chapter 11](#), page 306 to begin.

VMware Workstation Pro: A hosted hypervisor by the VMware Corporation for Windows and Linux systems. Like with VMware Fusion Pro, we specifically need the pro edition of VMware Workstation. Check out the sidebar below, *Why Are We Going Pro?* If you'd like to know more. Otherwise, jump to [chapter 12](#), page 405.

VMware vSphere Hypervisor (ESXi): A bare-metal hypervisor by the VMware corporation. It is very picky about hardware compatibility, but it is available for free, with an easy-to-use HTML5 web interface, compatible with most modern web browsers. Navigate to [chapter 13](#), page 515.

Why Are We Going Pro?

VMware Fusion has the "Fusion" and "Fusion Pro" editions, while VMware Workstation has "Workstation Player", and "Workstation Pro". The reason the professional edition of VMware Fusion or Workstation is required, is for the virtual network customization feature. This feature is available only in the professional editions of Fusion and Workstation, and we need it in order to create the lab network segments with the correct network connectivity limitations. We need to be absolutely sure that the lab network has only one path for network traffic to flow – through our pfSense firewall VM.

8.2 Hypervisor Guide – Chapter Outline

All of the hypervisor guides above follow the same general outline, described in detail below:

- Installing the Hypervisor
- Customizing the Hypervisor
- Step-by-step guide on creating the first Virtual Machine, pfSense
 - Performing resource allocations
 - Customizing virtual hardware
 - Installing the pfSense operating system
 - Performing initial pfSense configuration from the command-line
 - Assigning the WAN, LAN and OPT1 interfaces
 - Setting interface IP addresses
 - Defining DHCP scopes for the LAN and OPT1 networks
 - Accessing the Webconfigurator, and completing the initial setup wizard
 - Navigating to Chapter 14 (*pfSense Firewall Rules and Network Services*) and performing the remaining pfSense setup tasks
 - Coming back to the hypervisor setup chapter, and
 - Create an initial, baseline snapshot for the pfSense VM
- Coming back to the hypervisor setup chapter to create the SIEM, IPS, and Kali Virtual machines
 - Installing Ubuntu Server to the SIEM and IPS virtual machines
 - Installing Kali Linux to the Kali VM
 - Logging in via virtual machine console, testing network connectivity and installing software updates for SIEM, IPS and Kali VMs
- Installing Metasploitable 2 VM
 - Perform any necessary steps to register the Metasploitable 2 VM
 - Power on VM, log in via virtual console to verify functionality
 - Record MAC address and configure static DHCP mapping on pfSense VM
 - Remind students that this mapping won't work... yet.
- Create baseline snapshots for pfSense, SIEM, IPS, Kali, and Metasploitable 2 VMs
- Next Steps
 - Guide Students Towards chapters absolutely necessary to enable full lab functionality
 - *Chapter 15: Routing and Remote Access for Hosted Hypervisors*
 - *Chapter 16: Routing and Remote Access for Bare-Metal Hypervisors*
 - *Chapter 17: Network Intrusion Detection*
 - *Chapter 18: Setting Up Splunk*
 - Guide Students Towards Recommended Chapters (to make the lab even better)
 - *Chapter 19: End of the Beginning (Customization Options)*
 - *Chapter 20: Extra Credit (Extra lab features)*

Pick a hypervisor, get started, and I'll see you again at the finish line!

A note on repetitive content

Before we jump into the heart of this book, I would like to offer a final reminder: **This is not a book you will probably want or need to read from front to back, unless you plan on installing multiple VM labs on multiple hypervisors.** You're expected to jump to the content you need for a chosen hypervisor, then the required and recommended content to make your lab fully functional and more secure. This means that this book has a lot of steps in the various hypervisor setup chapters that are nearly identical. I assure you that I took every single effort that I could to be lazy, deduplicate efforts, and simplify instructions wherever I could. However, each of these hypervisors has different configuration options, menus, and verbiage for performing the same tasks, and I am nothing if not thorough because I believe students both need and deserve constant re-assurance and affirmation that they are performing steps correctly.

Full Accommodations

The chapter lessons and guides from here on out will mostly feature screen captures of various tasks being done on a Windows PC, except where instructions are specifically for MacOS or Linux. **The content is designed to where students can follow along, regardless of what desktop operating system they prefer to use.** Remember way back in chapter 1 where I gave you a pick list of software for Windows, MacOS, and Linux? There was a good reason for that! All users, regardless of preferred desktop OS will be accommodated and provided with the guidance necessary to succeed.

Chapter 9 Patch Notes

-Most of this chapter was built using Chapter 10 (Virtualbox) as a template. Key terminology changed when needed, installation processes, special caveats, etc. but by and far, the overall layout is nearly identical.

-Installation steps and system requirements go a lot further in-depth, and cross-reference back to chapter 5 quite a bit. Introduced `msinfo32` as an alternative to `systeminfo` for gathering system data and confirming that Hyper-V requirements have been met.

-Added some notes on Hyper-V not playing well with other hypervisors, and downright refusing to install if it finds another hypervisor running.

-Introduced students to the `optionalfeatures` run prompt option, instead of having to guess where Microsoft hid the *Turn Windows Features On or Off* option in the control panel this time around.

-Microsoft introduced the *Default Virtual Switch*. It basically operates as a NAT network segment. So, I talked about that, if students need it to deal with weird network problems.

-When creating virtual machines via the New Virtual Machine Wizard, I advise students to create their VMs and store them in a custom subdirectory underneath the *Virtual Machines* directory (*Hyper-V Settings*), in order to make for better housekeeping. I recall older versions Client Hyper-V dumping all the VM config files in one directory with UUIDs for file/folder names. This at least maintains some semblance of organization and sanity.

-Discussed Gen1 vs Gen 2 virtual machines a little bit more. TLDR: use Gen1 VMs to avoid compatibility problems with Linux and FreeBSD VMs.

-Discovered that *Dynamic Memory*, at some point and time, was terribly broken with Linux/BSD virtual machines, so now I explicitly advise against using it.

-Somewhere between now, and the first book the automatic checkpoints feature was introduced. I advise students to disable them to save space, and later tell them that if they have space, to enable them if that's what they really want.

-Explained to students that Hyper-V is somewhat lazy and waits until first boot to assign MAC addresses to virtual machine NICs.

-Advised students to start doing asset management and document VM settings. Provided an example entry and a template for them to do this

-Updated the installation instructions to teach users how to manually point configure/troubleshoot proxy settings the apt package manager on SIEM, IPS, and Kali. For the SIEM and IPS VMs, is a more of a here's how to fix this, if it was configured incorrectly. For the Kali users... something happened when Kali Linux 2020.3 came out, and the installer no longer asks if the users would like to configure an HTTP proxy anymore, so here I am, cleaning up the trash once again.

-Speaking of taking out the trash... the Kali Linux Installer Menu is bugged with Client Hyper-V. I have no idea how long this has been an issue, but I documented work-rounds and submitted a bug.

-Included a note to ensure port mirroring and MAC address spoofing is properly configured at the end of the chapter. Backported this change to chapter 10 for promiscuous mode configuration.

Chapter 9: Client Hyper-V

Client Hyper-V is a hypervisor created by Microsoft designed to run on Windows 8.1 and Windows 10 desktop operating systems. I have a personal preference towards Windows as my desktop operating system of choice (mainly because I am avid PC gamer). Discovering that Microsoft not only provides a free hypervisor, but that it has set of features comparable to other commercial hypervisors is just a nice bonus. Client Hyper-V is *technically* a bare-metal hypervisor, but I think of it as a bare-metal hypervisor that behaves an awful lot like a hosted hypervisor.

As one might guess, Client Hyper-V isn't the only version of the hypervisor available. Much like VMWare ESXi, Microsoft has made Hyper-V available as dedicated bare-metal server (Microsoft Hyper-V Server). Additionally, the hypervisor can be installed as a "Role" on the various versions of Windows Server operating systems. As a reminder, this chapter focuses on how to install and configure Client Hyper-V on Windows 10, though these instructions should be compatible with Windows 8.1.

9.1 Prerequisites

Before we begin the installation process, there are few prerequisites that students must confirm. While Client Hyper-V is a free feature, it is only available on certain versions of Windows 10 and Windows 8.1:

- Windows 8.1 Pro
- Windows 8.1 Enterprise
- Windows 10 Education
- Windows 10 Pro
- Windows 10 Enterprise

Note: Recall in Chapter 5 (p. 55, *64-bit support (Yes, we're having this conversation)*) that 64-bit hardware and operating system support should be a given. That means you need one of the 64-bit versions of the operating systems listed above.

Additionally, while it isn't specifically called out as a requirement, the disk in which you will be storing and running your virtual machines must be formatted with the NTFS file system. This is another one of those requirements (like 64-bit support), that should be a non-issue – every major Windows release since forever utilizes NTFS by default, even on SSDs. Even so, this is a problem I ran into when writing the first edition of this book, so if you experience problems with your virtual machines failing to initialize, double check that the disk NTFS formatted.

There are additional specific features Client Hyper-V requires:

Requirement	In plain English
VM Monitor Mode Extensions	"Your CPU must support AMD-V or Intel VT-x"
Virtualization Enabled in Firmware	"Does the motherboard support AMD-V or VT-x? Is it enabled in the BIOS?"
Second Level Address Translation	Also known as "Nested Paging". A method of bypassing some of the memory management overhead required for running virtual machines. There's a very good chance that if the CPU supports AMD-V or VT-x, that it also has this feature.
Data Execution Prevention Available	A security feature built into Windows, as well as a hardware feature of most modern CPUs. Sometimes called "Execute Disable Bit", "NX-bit", or "Enhanced Virus Protection". Practically every modern Intel/AMD processor supports this feature, and Windows enables it by default.

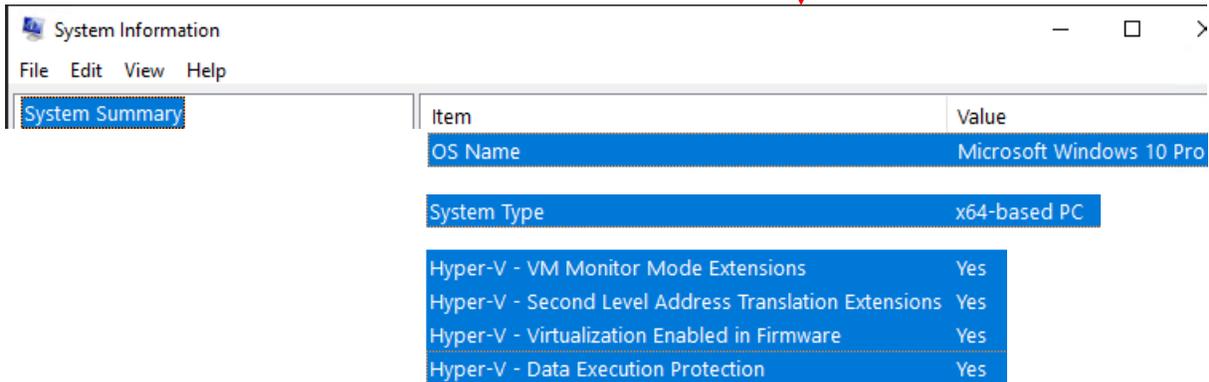
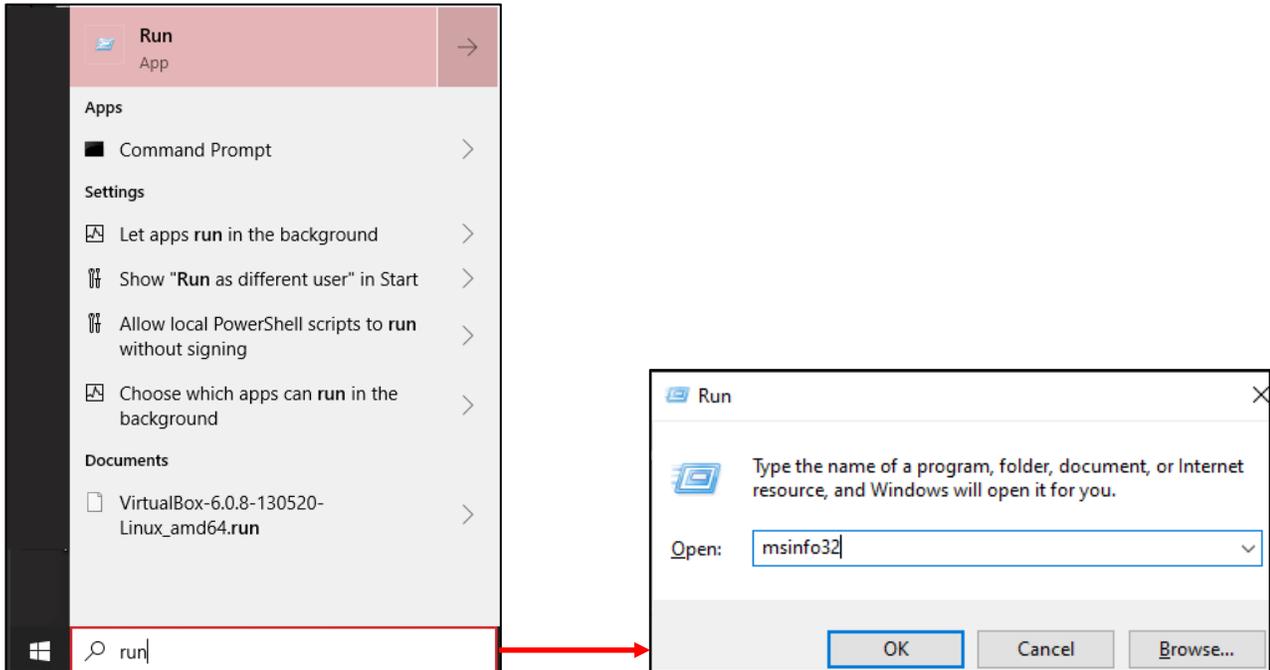
9.1.2 msinfo32

The fastest way of checking the version of Windows installed, and whether or not the system meets the necessary Hyper-V requirements is through the `msinfo32` command. To access this command, click the *Start* button, type `run` into the search bar, select the *run* prompt option, type in `msinfo32` in the *Open* input box, then click *OK*, or hit *enter*.

A window titled *System Information* appears, that is split into two panes. By default, the text *System Summary* should be highlighted on the left pane. If it is not, left-click on it to highlight it. On the right pane, a slew of information, divided into two columns titled *Item* and *Value* appear. The *OS Name* item will display what version of Windows 8 or Windows 10 is running on the system. The *System Type* item will read *x64-based PC* if it supports 64-bit operation. Scroll to the bottom of the right pane, and there will be four items labeled:

Hyper-V – VM Monitor Mode Extensions
Hyper-V – Second Level Address Translation Extensions
Hyper-V – Virtualization Enabled in Firmware
Hyper- V – Data Execution Protection

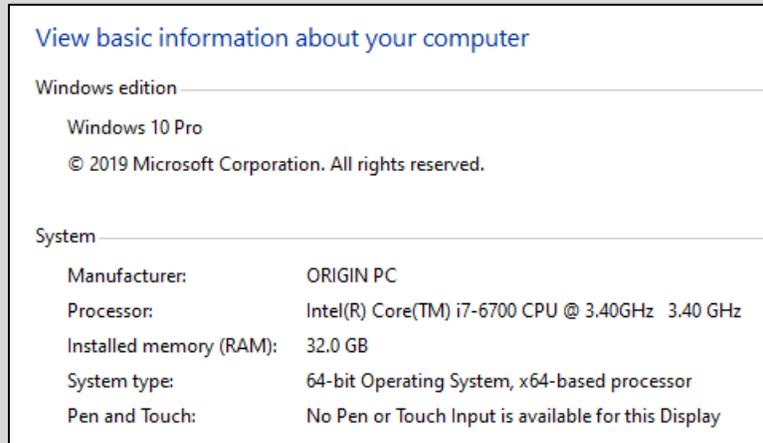
All of these items must have a value of *Yes* in order to successfully install Client Hyper-V.



9-1: Click the Start button and type run to bring up the *Run* app. In the *Run* window, type msinfo32 in the *Open* input box, then hit enter, or click *OK*. Under *System Summary*, check the *OS Name* item, and compare the *Value* to the supported Windows releases identified in [section 9.1](#) (p. 86). Check the *System Type* item and confirm the *Value* reads *x64-based PC*. Finally, check all four items labeled *Hyper-V – VM Monitor Mode Extensions*, *Hyper-V – Second Level Address Translation Extensions*, *Hyper-V – Virtualization Enabled in Firmware* and *Hyper-V – Data Execution Protection*. Ensure the *Value* of all four items reads *Yes*.

We Have Ways

There's more than one way to acquire the information above. For example, holding the Windows (or Meta) key while pressing the pause/break key will bring up a window titled *System*. Under the section *View Basic Information about your computer*, there are two subsections labeled *Windows Edition*, and *System*. The *Windows Edition* section will tell users what version of Windows is installed, while under *System*, the field labeled *System type* will confirm if the hardware and operating system provide 64-bit support.

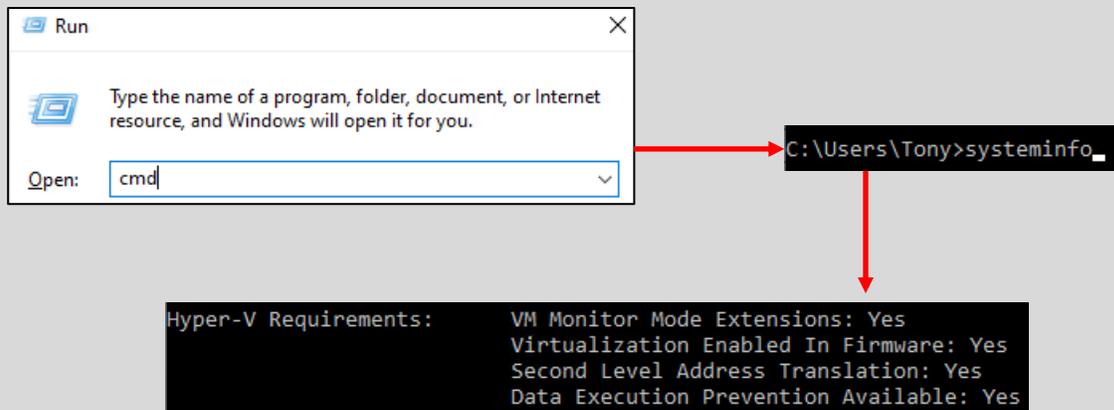


9-2: You can access this information by using the keyboard combo, Windows key + pause/break.

Next up, hold down the Windows key again, and this time, hit the 'r' key. This instantly brings up the *Run* prompt. Type `cmd` into the *Open* input box, and hit enter. The windows command line interface will appear. Type in the `systeminfo`. Think of this command as a command-line version of `msinfo32`. The only output we need is the Section labeled *Hyper-V Requirements*. And just like before, there are four values:

VM Monitor Mode Extensions
Virtualization Enabled in Firmware
Second Level Address Translation
Data Execution Prevention Available

Same as with `msinfo32`, all of these items must read *Yes* in order to successfully install Client Hyper-V.



9-3: The Windows key + r keyboard shortcut is a quick way of accessing the *Run* prompt. Type in `cmd` and hit enter to open the Windows command prompt. In the command prompt, type `systeminfo` and hit enter. This command will gather the exact same information as `msinfo32`, and display it in the command prompt window. The only section we're interested in is the four values under *Hyper-V Requirements*. Just like with `msinfo32`, all four of them need to read *Yes* in order to install Client Hyper-V.

Troubleshooting Prerequisites

Let's say you know for certain that your CPU has all of the right hardware requirements and extensions for Hyper-V (e.g., 64-bit support, AMD-V or VT-x, Nested Paging, DEP/NX-bit, etc.), but `msinfo32` or `systeminfo` are refusing to acknowledge those features are there. The only advice I can offer you is to check the BIOS/UEFI settings for your system's motherboard. It's one thing if the hardware supports the necessary features, but it's an entirely different problem confirming that those features are both supported and enabled.

We briefly touched on this topic in Chapter 5, [section 5.4](#) (pp. 54-55), but if you are using a pre-built system (e.g., Dell, HP, Lenovo, Acer, etc.), look up your system's model number and documentation to see if the BIOS has virtualization support. Otherwise, if you're using a custom-built system, you'll want to refer to the documentation that came with your motherboard to see if virtualization support is offered, and how to enable it.

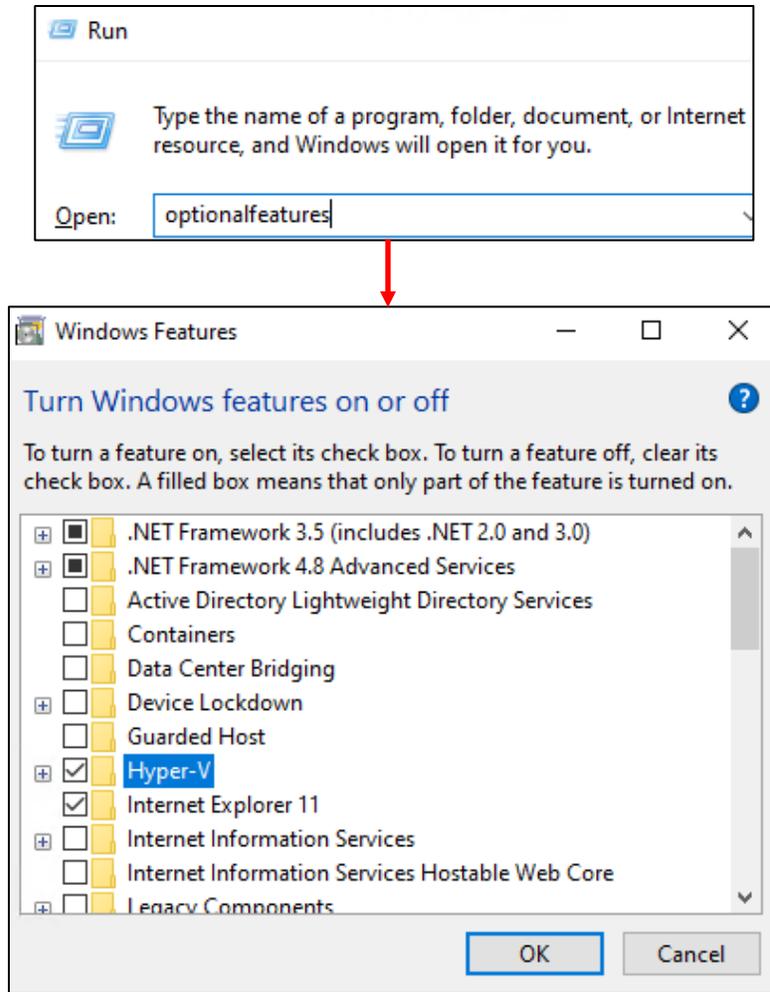
Some students may get the following error from `systeminfo` or `msinfo32`:

```
A hypervisor has been detected. Features required for Hyper-V will not be displayed.
```

If you encountered this error, it means that Windows detected that another hypervisor has been installed on your system. Unfortunately, Client Hyper-V does not play well with others, and requires other hypervisors to be uninstalled. If you're getting this error, and have no other hypervisors installed, another possibility is that Client Hyper-V may already be installed. Check to see if the *Hyper-V Manager* application is available on your system.

9.2 Installing Client Hyper-V

Now that students have confirmed their system meets the necessary requirements, the next step is installing Client Hyper-V, and all of its components. Begin by opening the run prompt, and typing `optionalfeatures` in the *Open* input box. A window titled *Windows Features* will appear, with the description, *Turn Windows features on or off*. Click the checkbox next to the feature labeled *Hyper-V*, then click *OK*. Windows will handle the rest. Please be aware that the system will reboot as a part of the installation process.



9-4: The installation process is very simple. Open the *Run* prompt (start menu, search for run, or use the Windows key + 'r' keyboard shortcut), type in `optionalfeatures` then hit enter. Click the *Hyper-V* checkbox, then click *OK*. The system will reboot as a part of the installation process. Students may also lose network connectivity, but we'll be fixing that in just a moment.

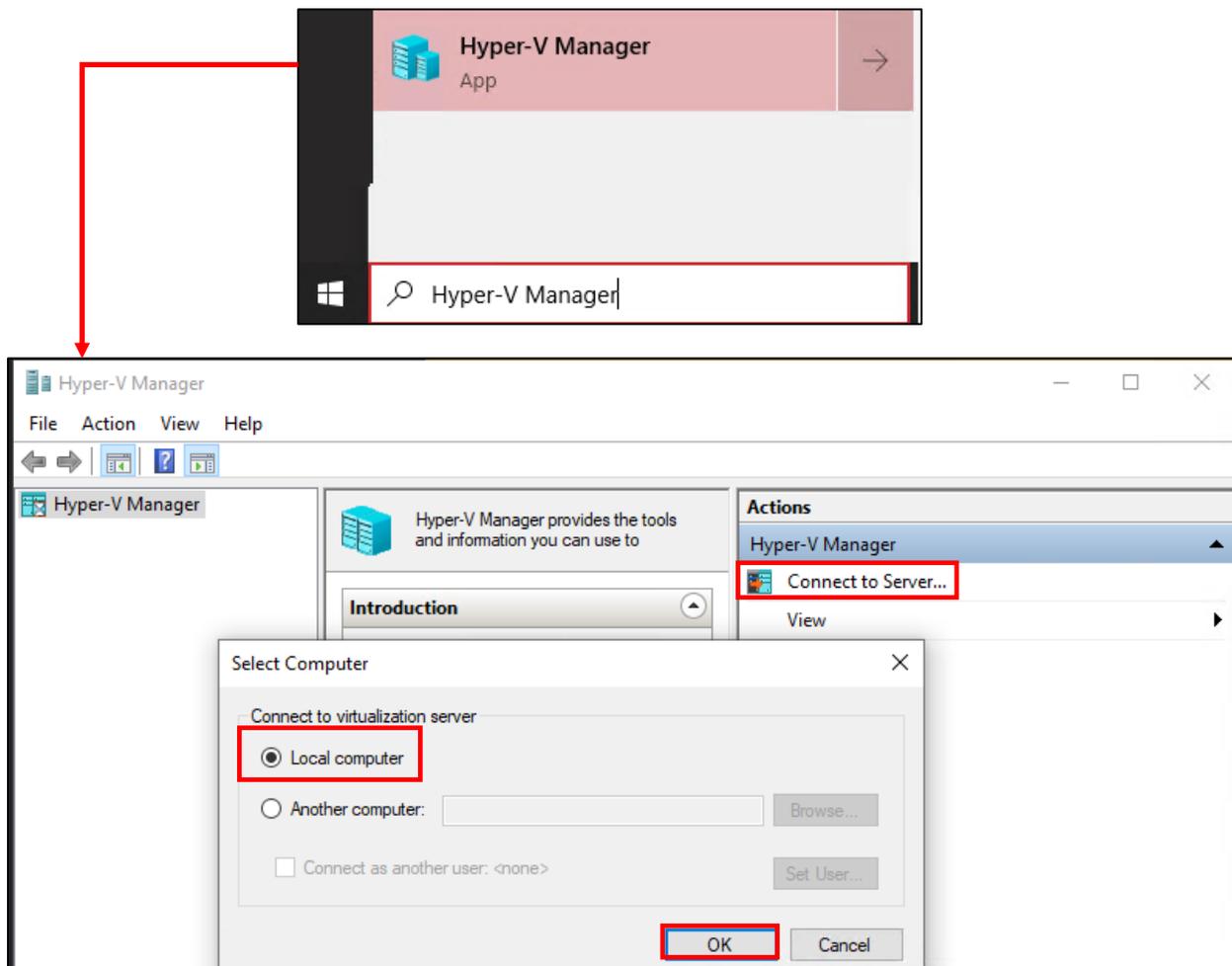
9.3 Customizing Client Hyper-V

With Client Hyper-V installed, students will need to become familiar with *Hyper-V Manager*, the application used to configure the hypervisor. In the next two sections, we will be covering the *Hyper-V Settings menu*, as well as the *Virtual Switch Manager*.

9.3.1 Hyper-V Settings

To begin, click the *start* button on the Windows desktop, type in *Hyper-V Manager* to locate the application, then click on it to start it up.

A window labeled *Hyper-V Manager* will appear, along with a pop-up window should labeled *Select Computer*. If for some reason this pop-up does not appear, click the *Connect to Server* option in right pane labeled *Actions*. Select the radio button labeled *Local computer*, then click the *OK* button to close the pop-up, and begin configuring the hypervisor.



9-5: Using the start menu, search for and start the *Hyper-V Manager* application. Students should be immediately be greeted by a pop-up labeled *Select Computer*. If not, click the *Connect to Server* option under the *Actions* pane. Click the *Local computer* radio button, then click *OK*.

The left pane updates with the name of the local Windows system. For example, my computer's hostname is STARFALL. A set of window panels in the middle pane labeled *Virtual Machines*, *Checkpoints*, and *Details* will appear, all of which are currently empty. In the right pane is an entire array of management actions that can be performed against Client Hyper-V. For now, locate the option *Hyper-V Settings* under the *Actions* pane, and click on it to open a new window labeled *Hyper-V Settings for [hostname]*. There is a large selection of settings available for Hyper-V, but here are the ones students should be aware of:

Virtual Hard Disks: When creating a virtual machine, Client Hyper-V allocates disk space on the hypervisor host in the form of .VHD or .VHDX (Virtual Hard Disk/Virtual Hard Disk Extended) files. This setting defines where Hyper-V will store these files. By default, Client Hyper-V will place these files in C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\. To change this setting, left click on the Virtual Hard Disks option to highlight it, then on the right pane, click the *Browse* button to choose a new folder. For example, on my system I store my VHD/VHDX files on E:\VM-Disks\.

Virtual Machines: In addition to VHD/VHDX files, Hyper-V virtual machines also have configuration files that define the various parameters for virtual machines. Like the *Virtual Hard Disks* option above, this setting can be used to store those configuration files in a specific location. The default location is C:\ProgramData\Microsoft\Windows\Hyper-V\, and can be changed by highlighting the *Virtual Machines* option, then clicking *Browse* on the right pane. On my system, I changed this setting to E:\VM\Configs\.

Enhanced Session Mode Policy: Later in this chapter after students create their virtual machines, they will need to use a virtual console in order to interact with their virtual machines. Hyper-V calls their virtual console *Virtual Machine Connection*. Enhanced Session Mode is a component allows the host computer (e.g., the hypervisor host) to share connected devices (printers, storage drives, etc.) with the virtual machines.

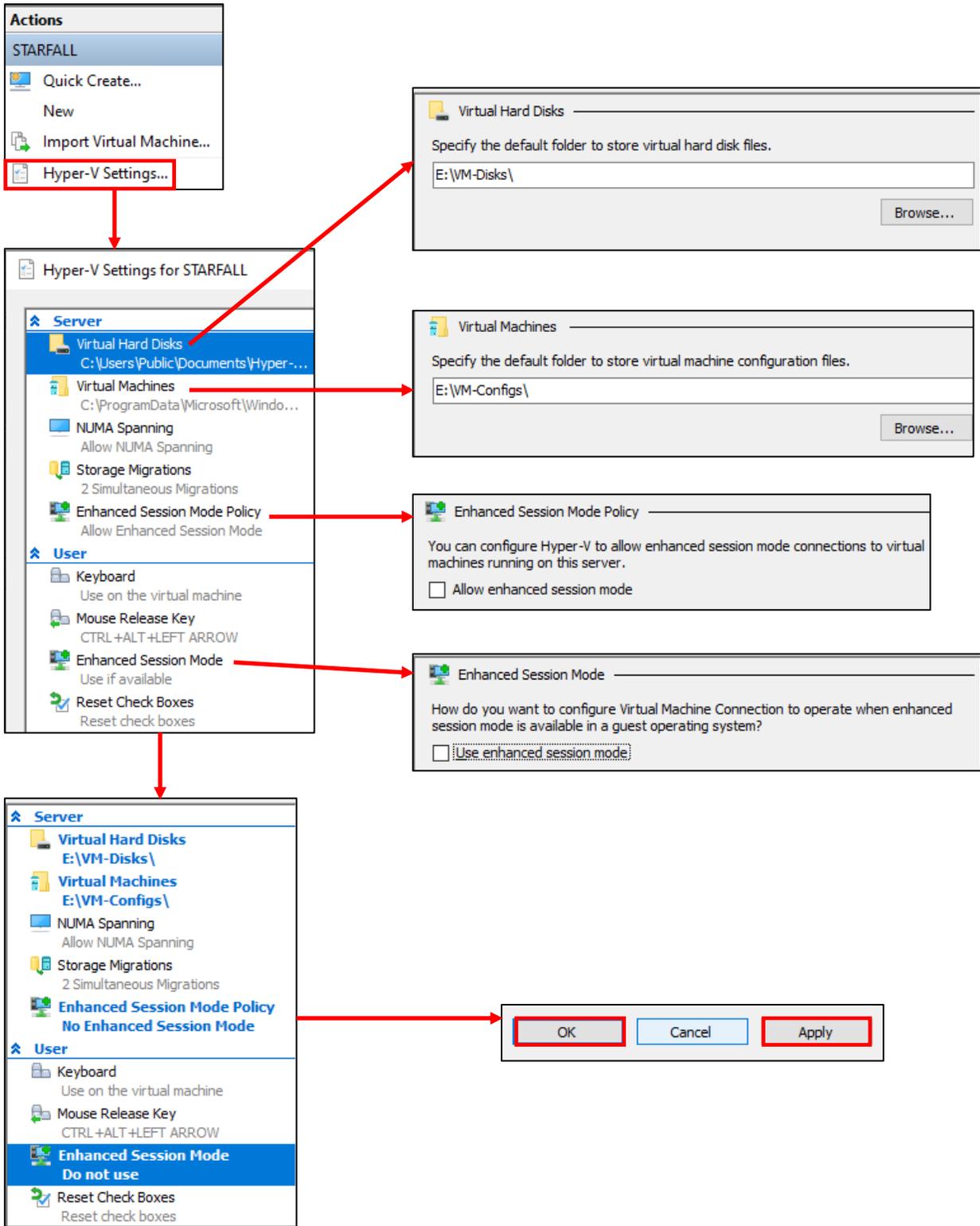
As students might have guessed, this feature is a risk that can affect the security of the lab environment, as well as separation between the hypervisor host and the virtual machines. As such, I highly recommend disabling this feature. To do so, click on the *Enhanced Session Mode Policy* option to highlight it, and uncheck the *Allow enhanced session mode* checkbox.

Mouse Release Key: As mentioned above, in order to interact with virtual machines, the hypervisor provides a virtual console to emulate a monitor, keyboard, and mouse. Clicking on a virtual console and bringing it into focus allows the virtual machine to "take over" the keyboard and mouse, while the *Mouse Release Key* combination is a keyboard shortcut that tells the virtual machine to relinquish control of the mouse back to the hypervisor host. With the mouse released, the user can click on another application window (or even the desktop itself) in order to regain control of the keyboard as well. There's no need to alter this setting, however, students should be aware that the default key combination is CTRL+ALT+LEFT ARROW, which translates into "hold down the ctrl and the alt keys, then press the left arrow key on the arrow pad."

Enhanced Session Mode: under the *Mouse Release Key* option is another setting titled *Enhanced Session Mode*. Note that this is different from the *Enhanced Session Mode Policy* setting above. *Enhanced Session Mode Policy* tells the hypervisor to never allow enhanced session mode connections, while the *Enhanced Session Mode* setting under users determines whether or not the *Virtual Machine Connection* (e.g., the Hyper-V virtual console) will request the feature.

Think of these two settings as "The hypervisor will provide Enhanced Sessions if the VM supports it" (*Enhanced Session Mode Policy*), vs. "The virtual console will request Enhanced Sessions by default" (*Enhanced Session Mode*). I recommend disabling this option. To do so, click on *Enhanced Session Mode* to highlight it, then uncheck the *Use enhanced session mode checkbox*.

When finished editing these settings, Click the *Apply* button in the lower right corner, followed by the *OK* button to close the *Hyper-V Settings for [hostname]* window.



9-6: Open up the *Hyper-V Settings* menu from the *Actions* pane in the *Hyper-V Manager*. Adjust the *Virtual Hard Disks* and *Virtual Machines* settings as necessary to reflect the hypervisor host's storage configuration. Make note of the *Mouse Release Key* combination, and be sure to disable both the *Enhanced Session Mode*, and *Enhanced Session Mode Policy*. Changed items will be highlighted with blue text. Click *Apply*, then *OK* to apply these settings, then close the *Hyper-V Settings* menu.

9.3.2 Virtual Switch Manager

Under the *Actions* pane in the *Hyper-V Manager* window, click on *Virtual Switch Manager*. A new window titled *Virtual Switch Manager for [hostname]* appears. While Hyper-V utilizes virtual switches like most bare-metal hypervisors, Hyper-V features 3 different types of virtual switches, each with unique capabilities, sort of like how there are different types of network segments on most hosted hypervisors. As a general reminder, students may review the different types of virtual network connectivity in [Chapter 4](#) (pp. 44-50).

Just like with the *Hyper-V Settings* menu, there are two panes to the *Virtual Switch Manager* menu. The left pane is used for navigation, while the right pane shows configuration details for the selected option on the left pane. The option *New virtual network switch* should be highlighted by default. If it's not, left-click to highlight it, and display the *Create virtual switch options* on the right pane. Hyper-V has three different types of virtual switches users can create:

External: Operates identically to a bridged virtual network segment. When users create an external switch, users are then asked to select which physical network card on the system the external switch will "bind" or bridge to.

Internal: Identical to host-only network segments, internal virtual switches allow all VMs attached to the virtual switch to communicate with one another, as well as with the hypervisor host, through the creation of a virtual network interface.

Private: These virtual switches are nearly the same as internal virtual switches, however the key difference is that private virtual switches do not create a virtual network interface for the hypervisor host. Virtual machines on private virtual switches may only communicate with one another.

Students will need to create four virtual switches for our lab environment: One External to serve as our Bridged network segment that connects our lab environment to the physical network, One Internal for our Management network, and two Private virtual switches to act as the IPS 1 and IPS 2 network segments.

Under the window titled *What type of virtual switch do you want to create*, left click *External* to highlight it, then click the *Create Virtual Switch* button. A new virtual switch appears on the left navigation pane, helpfully named *New Virtual Switch*. This new entry will be highlighted, and the right pane will display *Virtual Switch Properties* for this newly created virtual switch.

To begin, select the *Name* input box, and change the name of the virtual switch to something a little more descriptive, such as *Bridged Virtual Switch*. Next, under the section labeled *Connection type*, is the question *What do you to connect this virtual switch to?* Ensure that the *External* network radio button is selected.

This radio button features a drop-down menu that is a list of physical network interfaces installed on the host operating system. On my laptop for example, I had a choice of the following interfaces:

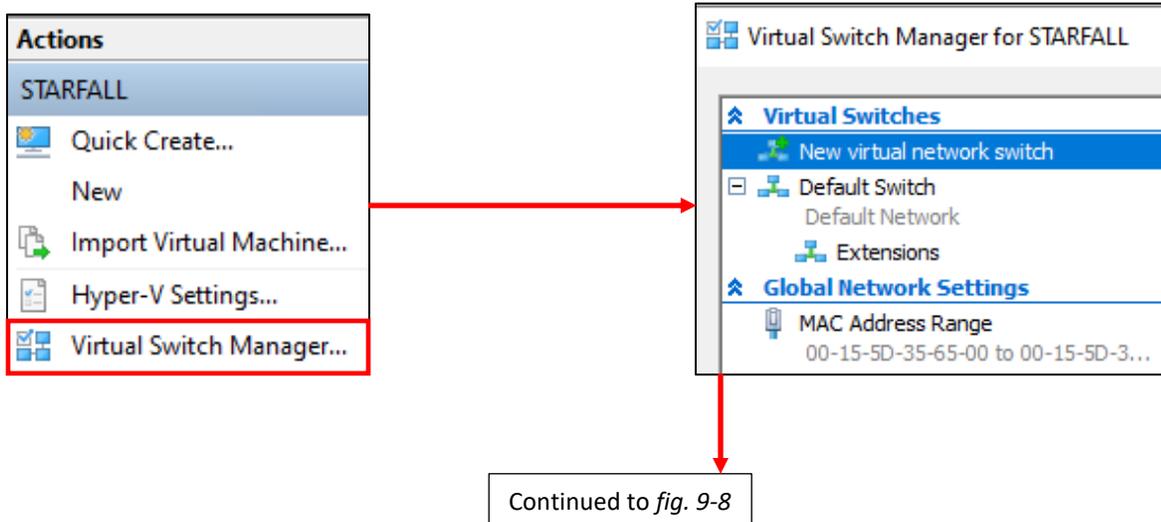
Killer E2400 Gigabit Ethernet Controller
Intel® Dual Band Wireless-AC 7265

Students should ensure that the network interface they use for connecting to their local network and/or the internet is selected. Using my local network as an example again, I prefer used wired network connectivity, so I selected the Killer gigabit ethernet controller. Underneath the drop-down is a checkbox labeled *Allow management operating system to share this network adapter*. **Make sure this option is checked.**

Click on the *New virtual network switch* option on the left pane again, and this time, select *Internal* as the type of virtual switch to create. When the *Virtual Switch Properties* pane appears, change the *Name* input box to Management Virtual Switch.

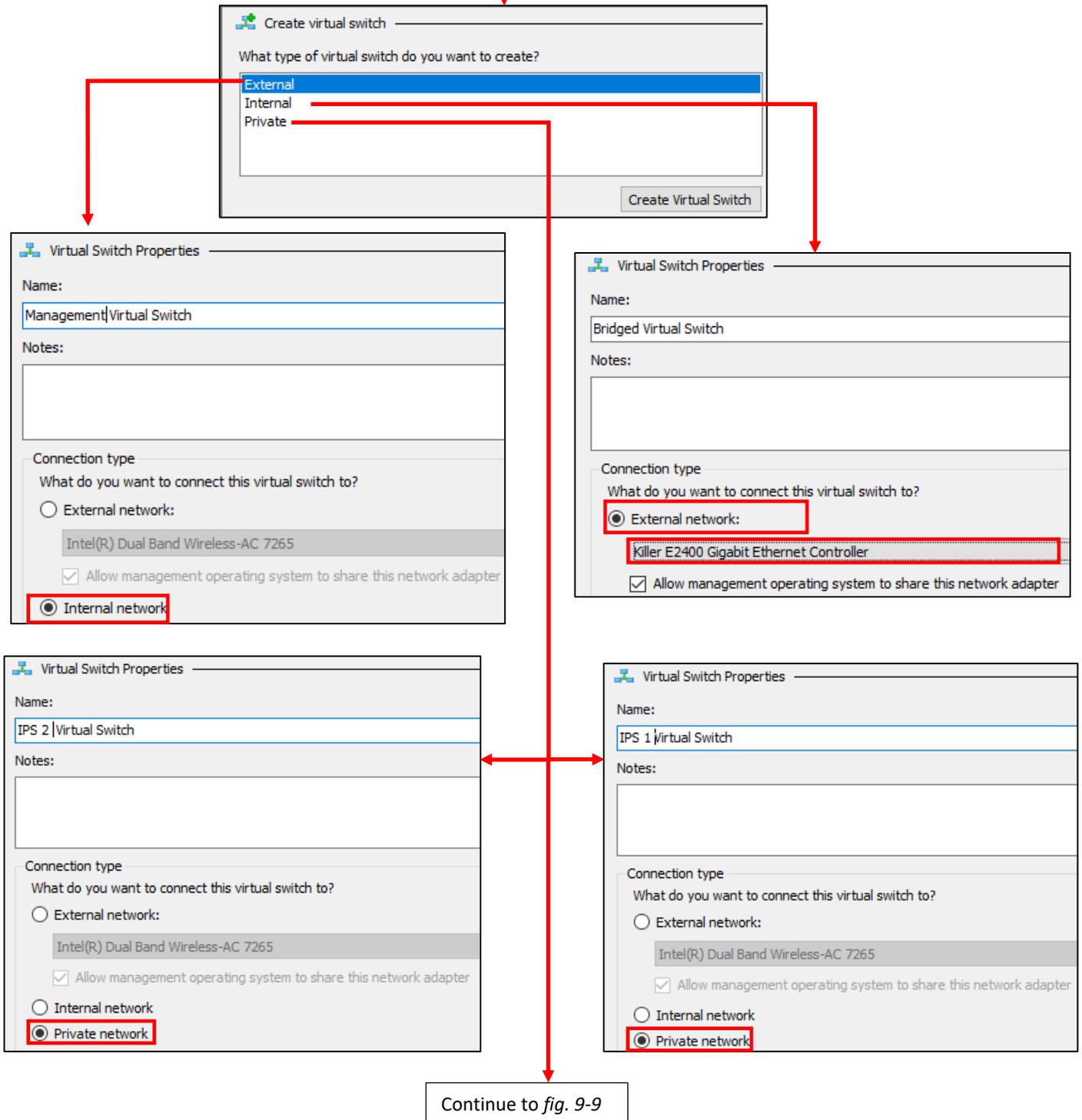
Afterwards click on the *New virtual network switch* option on the left pane once more, selecting *Private* as the virtual switch type. Change the *Name* of this virtual switch to IPS 1 Virtual Switch. Click *New virtual network switch* one last time, create one more *Private* virtual switch, and change its *Name* to IPS 2 Virtual Switch.

After creating all four virtual switches, click the *Apply* button in the bottom right corner of the window, then *OK* to close the *Virtual Switch Manager* menu.

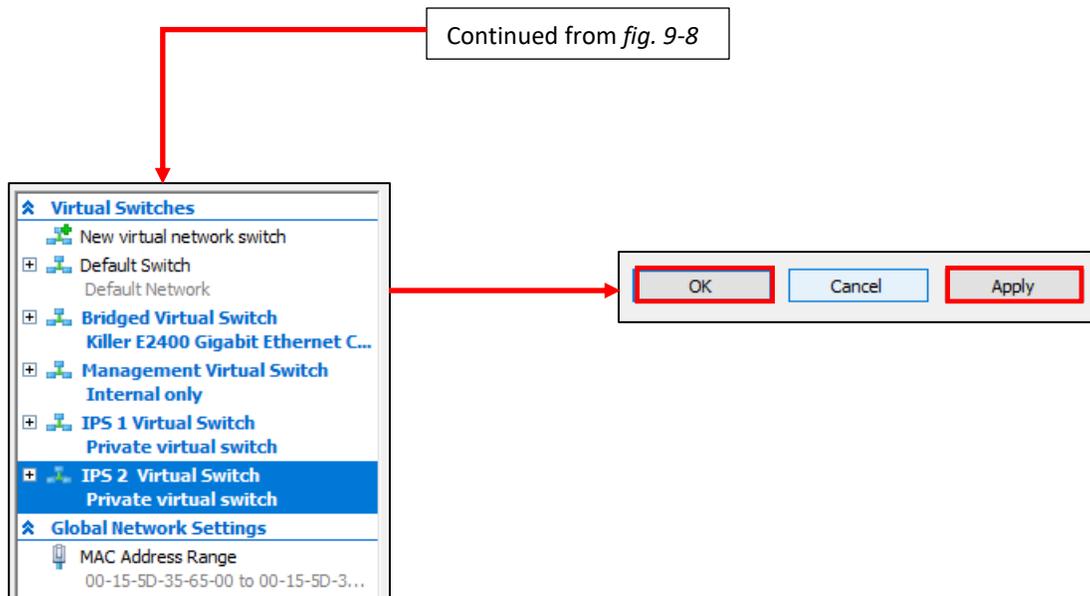


9-7: In the *Hyper-V Manager* window, select *Virtual Switch Manager* from the *Actions* pane. In the new window that pops up, click the *New virtual network switch* option from the left pane.

Continued from *fig. 9-7*



9-8: Students are presented with three different types of virtual switches they can create. Create one *External* virtual switch named *Bridged Virtual Switch*. In the drop-down under the *External* radio button, be sure to select the network card used to connect to the local network and/or the internet. Additionally, I recommend checking the *Allow management operating system to share this network adapter* checkbox under the drop-down. Afterwards, create an *Internal* virtual switch named *Management Virtual Switch*, followed by two *Private* virtual switches, named *IPS 1 Virtual Switch* and *IPS 2 Virtual Switch*.



9-9: When finished, the left pane of the *Virtual Switch Manager* window will look similar to the illustration above. Click *Apply*, then *OK* to create these virtual switches, then close the menu.

Flip the Default Switch

Some students might have noticed that in addition to the four switches created in the section above, that there's another switch listed in the *Virtual Switch Manager* labeled *Default Switch*. When clicked on, you can view its properties like the other virtual switches, however it cannot be modified, or deleted. In small black text at the bottom of its Virtual Switch Properties, there is text that reads:

The Default Network switch automatically gives virtual machines access to the computer's network using NAT (network address translation).

In layman's terms, this virtual switch, which cannot be modified or deleted, operates like the NAT network segment of a hosted hypervisor. As mentioned in chapter 4, depending on how your local network is configured, it may be necessary to use a NAT network segment as opposed to a Bridged (or in the case of Hyper-V, External) network segment.

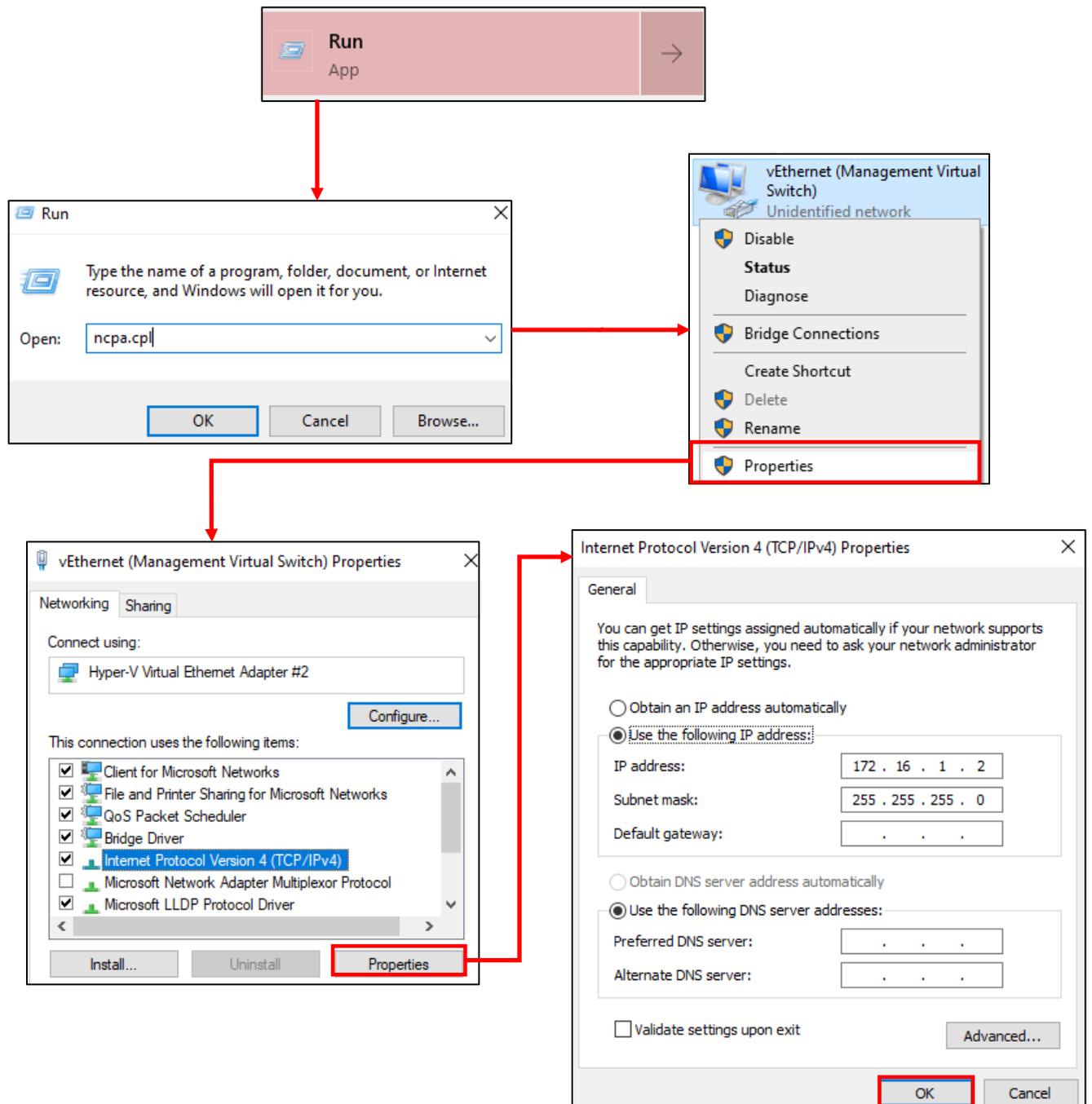
9.3.3 Configuring the Host-Only Network Interface (Management Virtual Switch)

In the previous section, students created an *Internal* virtual switch named *Management Virtual Switch*. This results in creating a virtual network interface on the hypervisor host labeled *vEthernet (Management Virtual Switch)*. Later in this chapter, students will need to use their web browser to configure the pfSense virtual machine in our lab environment. In order to facilitate this, that virtual network interface will need to have an IP address and subnet mask configured.

Begin by opening the *Run* prompt (Students can refer to [section 9.2](#), page 91 for a refresher, if needed) and typing `ncpa.cpl` into the *Open* input box, then press enter. This will open up the *Network Connections* panel without the need to figure out where Microsoft hid it in the latest Windows 10 overhaul.

Locate the network adapter icon titled *vEthernet (Management Virtual Switch)*. right-click on it, and select *Properties*. This opens a new window titled *vEthernet (Management Virtual Switch) Properties*. In the box labeled, *This connection uses the following items*, locate the entry labeled *Internet Protocol Version 4 (TCP/IPv4)*, left-click to highlight it, then click the *Properties* button.

This opens a new window labeled *Internet Protocol Version 4 (TCP/IPv4) Properties*. In the *General* tab below, make sure that the *Use the following IP address* radio button is selected, as well as the *Use the following DNS server addresses* radio button. In the *IP address* input box enter 172.16.1.2, and in the *Subnet mask* input box, enter 255.255.255.0. **All of the other remaining input boxes should be left blank.** Click *OK* to apply these settings.



9-10: Open the *Run* prompt, then type `ncpa.cpl` and hit enter. This opens up the *Network Connections* panel. Find the interface named *vEthernet (Management Virtual Switch)*, right click on it and select *Properties*. Click on *Internet Protocol Version 4 (TCP/IPv4)* in the *This connection uses the following items* window to highlight it, then click *Properties*. Click the *Use the following IP address*, and *Use the following DNS server addresses* radio buttons. Enter 172.16.1.2 as the IP address, and 255.255.255.0 as the Subnet mask. ***Leave all the remaining fields blank***, then click *OK*.

9.4 Building the First VM, pfSense

The pfSense virtual machine is responsible for binding the entire lab environment together. It is a well-supported firewall distribution with amazing ease of use and functionality. pfSense is also very modular, featuring a system for adding on additional functionality through BSD's software package manager, pkg.

It is recommended for students to download all of the ISOs (e.g., pfSense, Ubuntu Server and Kali Linux), and pre-built virtual machines (e.g., Metasploitable 2) required for their lab environment in advance. Check out chapter 1, [section 1.5.4](#), p. 26 for download links. Additionally, students must decompress the pfSense installation ISO before attempting to boot from it. [Section 1.8](#), pp. 33-35 covers how to do this on Windows, using 7-Zip.

9.4.1 VM Creation

Open the *Hyper-V Manager*, and under the *Actions* pane, select *New*, then *Virtual Machine* in the sub-menu that appears. This opens a new window aptly titled, *New Virtual Machine Wizard*. Since this is the first time students will be running the wizard, they will be greeted by a screen titled *Before You Begin*. It states that users can click the *Finish* button, and the hypervisor will create a virtual machine with default values selected. Click the checkbox labeled *Do not show this page again*, then click *Next* to continue.

The next screen, *Specify Name and Location*, appears. Enter pfSense into the *Name* input box, then click the checkbox *Store the virtual machine in a different location*. Append the name of the virtual machine to the end of the *Location* input box. For example, my virtual machine files are stored in the directory `E:\VM-Configs\`. That path then becomes `E:\VM-Configs\pfSense`. The default directory of `C:\ProgramData\Microsoft\Windows\Hyper-V\` would then become the directory `C:\ProgramData\Microsoft\Windows\Hyper-V\pfSense`. When finished, click the *Next*.

All About Location

You're probably wondering: *Why are we specifying a custom location for the virtual machine files?* Well, that's because while Client Hyper-V allows users to set a custom Virtual Machine directory in the *Hyper-V Settings* menu ([section 9.3.1](#), pp. 92-95), the problem is that directory that will hold **all** of the configuration files for **all** of the virtual machines. Hyper-V isn't smart enough to name the files logically, or create a subdirectory under that default directory for each virtual machine in order to keep things neat and tidy. So, I'm having you do that manually during the *New Virtual Machine Wizard*. Of course, students are under absolutely no obligation to do this, but it will make backing up virtual machine files and/or doing migrations much easier in the future, so I highly recommend it. Keep this in mind a little bit later in the wizard, because there is a similar setting for defining where Hyper-V stores the virtual machine's VHDX file.

Next up is the *Specify Generation* screen. Hyper-V has this concept of generations for their virtual machines. Without going deep into detail, pfSense as an operating system is based off of FreeBSD. Hyper-V only supports running FreeBSD as a generation 1 virtual machine. Click the *Generation 1* radio button, then click *Next* to proceed.

Generations

Generation 2 VMs have a couple more bells and whistles than their generation 1 counterparts, but unfortunately Hyper-V is very selective about what operating systems can be generation 2 VMs. FreeBSD for example, is not supported on generation 2 virtual machines, while Debian Linux 8.x and above are supported. However, most users won't miss the extra features available on generation 2 virtual machines. So, when in doubt, click the *Generation 1* radio button on the *Specify Generation* screen, if you're not sure whether or not the operating system is supported. If you're curious and want to know more, Microsoft has a documentation page dedicated to explaining the different features, as well as what operating systems are supported on what generation of virtual machine here: <https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/Should-I-create-a-generation-1-or-2-virtual-machine-in-Hyper-V>

The next screen, *Assign Memory*, asks students to specify how much RAM to allocate to the virtual machine. Type 512 into the input box labeled *Startup memory*, and uncheck the *Use Dynamic Memory for this virtual machine* checkbox. This will allocate 512MB of RAM to the virtual machine. Click the *Next* button to proceed to the next screen.

Dynamic Entry

Dynamic Memory is a special feature that allows Hyper-V to allocate more or less memory to a virtual machine on the fly depending on its needs. In theory, it's a great idea to let the hypervisor handle memory allocations automatically, however I've been burned by this feature in the past – it doesn't exactly work correctly with some Linux virtual machines. The VM will successfully allocate more RAM as required, then refuse to give it back. This leads to the host system running out of memory and crashing, taking everything else down with it. In general, I recommend unchecking the *Use Dynamic Memory for this virtual machine* checkbox, and manually re-adjusting the amount of RAM allocated to your virtual machines as needed.

The *Configure Networking* screen features a drop-down labeled *Connection*. This drop-down will define which virtual switch the first network interface of this virtual machine will be connected to. Select *Bridged Virtual Switch*, then click the *Next* button.

The *Connect Virtual Hard Disk* screen allows students to create a virtual hard disk file, attach an already existing one, or create a virtual hard disk later. Click the *Create a virtual hard disk* radio button. There are three input boxes under this radio button. The first one is the *Name* field, used

to define the name of the .vhdx file. The default name of pfSense.vhdx should be fine. The second input box is *Location*, which defines where on the filesystem the virtual disk file will reside.

Recall in section 9.1.3 the *Virtual Hard Disks* setting that defines where Hyper-V will store VHD/VHDX files by default. Normally, that setting would be the location the wizard would use for storing the virtual disk, but if students selected the *Store the virtual machine in a different location* check box on the *Specify Name and Location* screen, Hyper-V will create sub-directories under that custom directory named [virtual machine name]\Virtual Hard Disks\. Confused? Let's go over some examples.

I chose to store the files for the pfSense virtual machine in E:\VM-Configs\pfsense\, so the hypervisor creates:

```
E:\VM-Configs\pfSense\pfSense\Virtual Hard Disks\
```

and places the pfSense.vhdx file there.

Likewise, if students decided to use C:\ProgramData\Microsoft\Windows\Hyper-V\pfSense, Hyper-V creates:

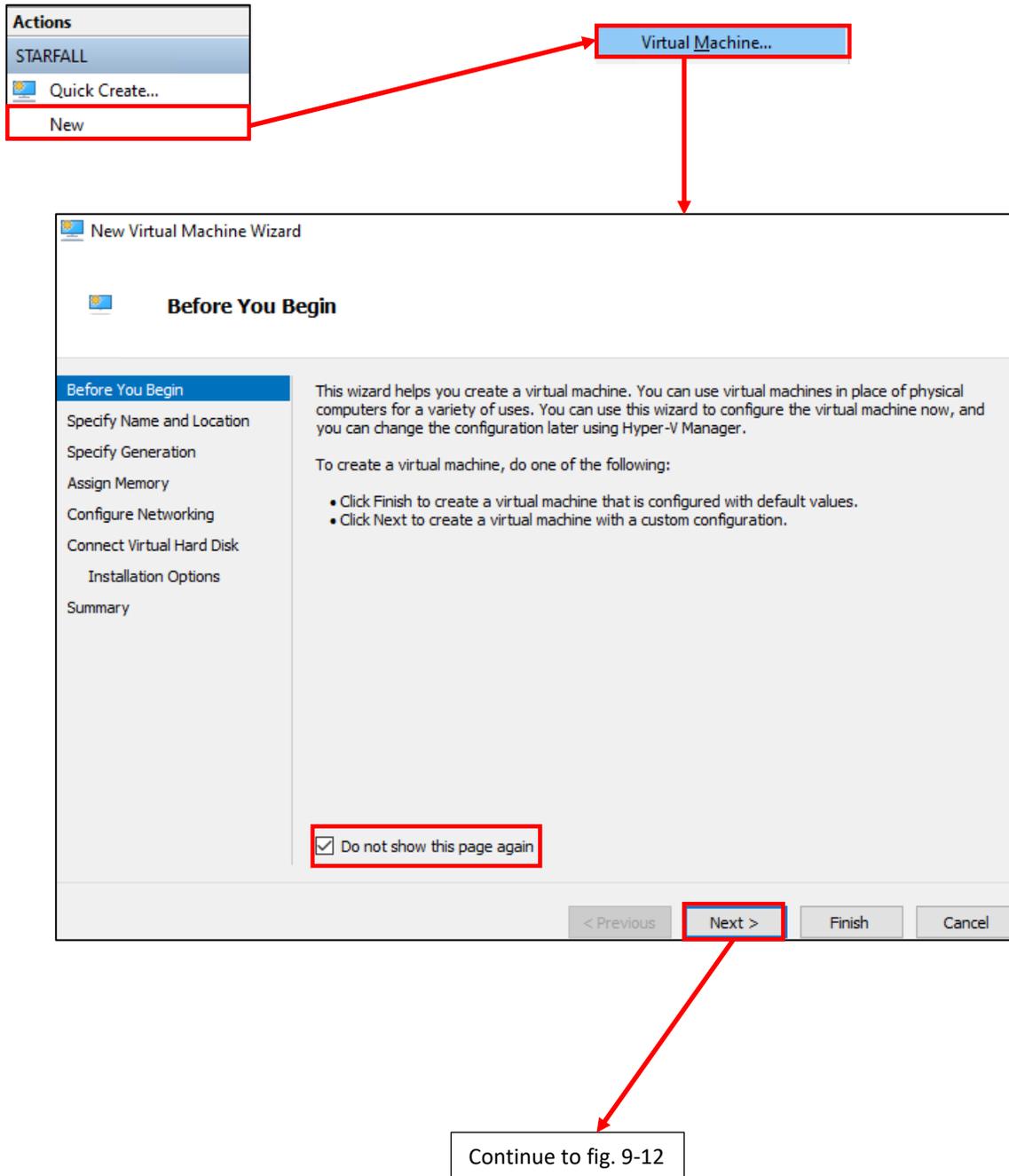
```
C:\ProgramData\Microsoft\Windows\Hyper-V\pfSense\pfSense\Virtual Hard Disks\
```

and stores pfSense.vhdx there.

With all of that out of the way, that leaves the *Size* input box. This setting defines how large the virtual hard disk will be. Change this setting from its default to 5, in order for Hyper-V to allocate 5GB of disk space for the pfSense virtual machine. Afterwards, click *Next* to continue.

The next screen is *Installation Options*. This screen allows users to tell the new virtual machine where it can find a bootable operating system, such as the pfSense installation ISO. Click the radio button labeled *Install an operating system from a bootable CD/DVD-ROM*, then in the *Media* section below, click the radio button labeled *Image file (.iso)*, then click the *Browse* button. Use the Windows file explorer to locate and select the decompressed pfSense ISO file. Once finished, click *Next* to proceed to the final screen.

The final screen in the wizard provides users with a brief summary of all the settings they have selected. Click the *Finish* button to create the pfSense VM.



9-11: Click *New* in the Hyper-V Manager's *Actions* pane, then click *Virtual Machine* in the sub-menu that appears. This starts the *New Virtual Machine Wizard*. If this is the first time students are running the wizard, The *Before You Begin* screen will appear. This first screen informs users that they can click the *Finish* button to create a virtual machine with default settings applied. Since our virtual machines are all going to vary in function and use, it is advised to not use this feature. Click the *Do not show this page again* checkbox, then click *Next* to proceed.

Continued from fig. 9-11

Specify Name and Location

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

Installation Options

Summary

Choose a name and location for this virtual machine.

The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.

Name:

You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.

Store the virtual machine in a different location

Location:

! If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.

Specify Generation

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

Installation Options

Summary

Choose the generation of this virtual machine.

Generation 1

This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.

Generation 2

This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.

! Once a virtual machine has been created, you cannot change its generation.

Assign Memory

Before You Begin

Specify Name and Location

Specify Generation

Assign Memory

Configure Networking

Connect Virtual Hard Disk

Installation Options

Summary

Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 12582912 MB. To improve performance, specify more than the minimum amount recommended for the operating system.

Startup memory: MB

Use Dynamic Memory for this virtual machine.

i When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.

Continued to fig. 9-13

9-12: Next up, the wizard asks for the name of the virtual machine, and where to store the virtual machine's configuration files. I recommend clicking the *Store the virtual machine in a different location* checkbox, and creating a subdirectory that shares the virtual machine's name. The next screen asks what generation of virtual machine to create. ***pfSense will not run in a Generation 2 virtual machine***, so be sure to select the *Generation 1* radio button. Next up, the wizard asks users to specify how much RAM (in Megabytes) to allocate to the virtual machine. The pfSense VM should have 512MB allocated. Additionally, ***it's extremely important to uncheck the Use Dynamic Memory for this virtual machine checkbox***.

Continued from fig. 9-12

Configure Networking

Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary

Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.

Connection: **Bridged Virtual Switch**

Connect Virtual Hard Disk

Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary

A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

Create a virtual hard disk
Use this option to create a VHDX dynamically expanding virtual hard disk.

Name: pfSense.vhdx
Location: E:\VM-Configs\pfSense\pfSense\Virtual Hard Disks\ Browse...
Size: 5 GB (Maximum: 64 TB)

Installation Options

Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary

You can install an operating system now if you have access to the setup media, or you can install it later.

Install an operating system later
 Install an operating system from a bootable CD/DVD-ROM

Media
 Physical CD/DVD drive: [Dropdown]
 Image file (.iso): D:\Tony\Desktop\lab_isos\pfSense-CE-2.4.5-REL Browse...

Continued to fig. 9-14

9-13: The next screen has users decide which virtual switch to connect the virtual machine's first network interface to. Select *Bridged Virtual Switch* from the drop-down menu labeled *Connection*. On the *Connect Virtual Hard Disk* screen, select the *Create a virtual hard disk* radio button. Accept the default *Name* and *Location*, but change the *Size* from its default down to 5GB. Next up on the *Installation Options* screen, select the *Install an operating system from a bootable CD/DVD-ROM* radio button, select the *Image file (.iso)* radio button, then locate and select the decompressed pfSense ISO.

Continued from fig. 9-13



Name	State	CPU Usage	Assigned Memory	Uptime	Status
pfsense	Off				

9-14: That brings students to the summary page of the virtual machine wizard. Click the *Finish* button to close the wizard, and create the pfSense virtual machine. The new VM should appear in the middle pane of the *Hyper-V Manager*, under the *Virtual Machines* section.

9.4.2 pfSense Virtual Machine Settings (Part 1)

With the virtual machine created, we're just about ready to install an operating system. Before proceeding, there are a handful of adjustments to make to the virtual machine, and its hardware. These adjustments are to ensure the VM does not have any extra functionality enabled that is not needed, while adding additional hardware, or functionality that the lab will require. This will make our virtual machines leaner, and reduce possible attack surface. **It is recommended to perform this process for every virtual machine created for the lab environment.**

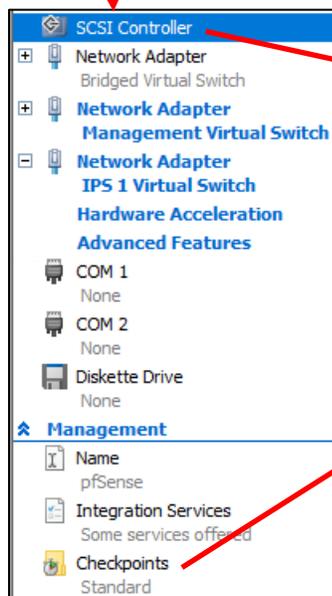
In the *Hyper-V Manager*, under the *Virtual Machines* section, right click on the pfSense VM entry, then select *Settings*. A new window titled *Settings for pfSense on [hostname]* will appear. Like most of the other Hyper-V menus, this screen is divided into a navigation pane on the left with items to select, and a details pane on the right with more information about the currently highlighted item.

The navigation pane is divided into a section labeled *Hardware*, and another section further below, labeled *Management*. By default, when this window is opened, The *Add Hardware* option should be selected, but if it isn't, click on it to highlight and select it. The right pane labeled, *Add Hardware* appears. This panel has a small window with a list of virtual hardware devices that can be added to the pfSense VM. Students will need to add two additional virtual network cards to the virtual machine. To do that, left click on the entry labeled *Network Adapter* to highlight/select it, then click the *Add* button. Immediately, the new network adapter is highlighted, and its settings appear in the right pane. The only setting students need to modify is the drop-down labeled *Virtual switch*. Select *Management Virtual Switch* from the drop-down, then click on the *Add Hardware* option on the navigation pane again.

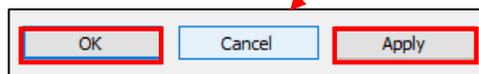
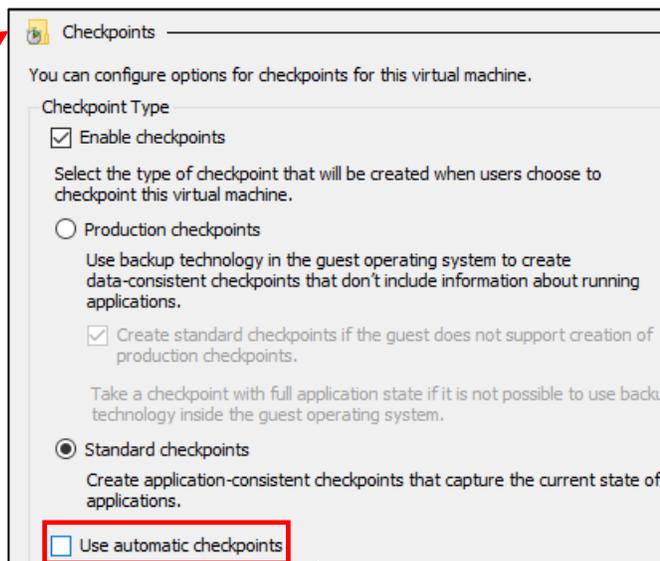
Once again, highlight the *Network Adapter* option, then click the *Add* button to create the third and final network adapter for the pfSense VM. This time around, select *IPS 1 Virtual Switch* from the *Virtual switch* drop-down on the *Network Adapter* settings pane. Click the + sign next to the highlighted network adapter on the left pane, and a set of options opens up underneath the network adapter connected to the IPS 1 virtual switch. Click the *Advanced Features* option, and the right pane updates to display an array of advanced networking options. Scroll down to the section labeled *Port mirroring*, and select the *Source* option from the *Mirroring mode* drop-down. **The mirror mode configuration is extremely important for the operation of both the Snort3 and Suricata IDS/IPS software, so please do not forget to perform this task.**

Next, select the *SCSI Controller* in the left pane, then click the *Remove* button on the right pane. If you did this correctly, the *SCSI Controller* item in the left pane will have ~~striketrough~~ text over it. Finally, select the *Checkpoints* option on the left pane. On the right pane, in the *Checkpoint Type* section, ensure that the *Enable checkpoints* checkbox is selected, followed by the *Standard checkpoints* radio button. Uncheck the *Use automatic checkpoints* checkbox. Afterwards, click the *Apply*, then *OK* button in the bottom right portion of the settings window to apply the changes we've made and close the pfSense VM settings menu.

Continued from fig. 9-15



To remove the SCSI controller from this virtual machine, click Remove. All virtual hard disks attached to this controller will be removed but not deleted.



9-16: Next up, remove the SCSI controller, since it isn't connecting anything, and finally, go to the *Checkpoints* option on the left pane. Under *Checkpoint Type*, ensure the *Enable checkpoints* checkbox is enabled, and the *Standard checkpoints* radio button is selected (they should be selected by default). Uncheck the *Use automatic checkpoints* checkbox. Once finished, click *Apply* then *OK* to apply the changes made, then close the virtual machine settings menu.

9.4.3 First Boot and OS Installation

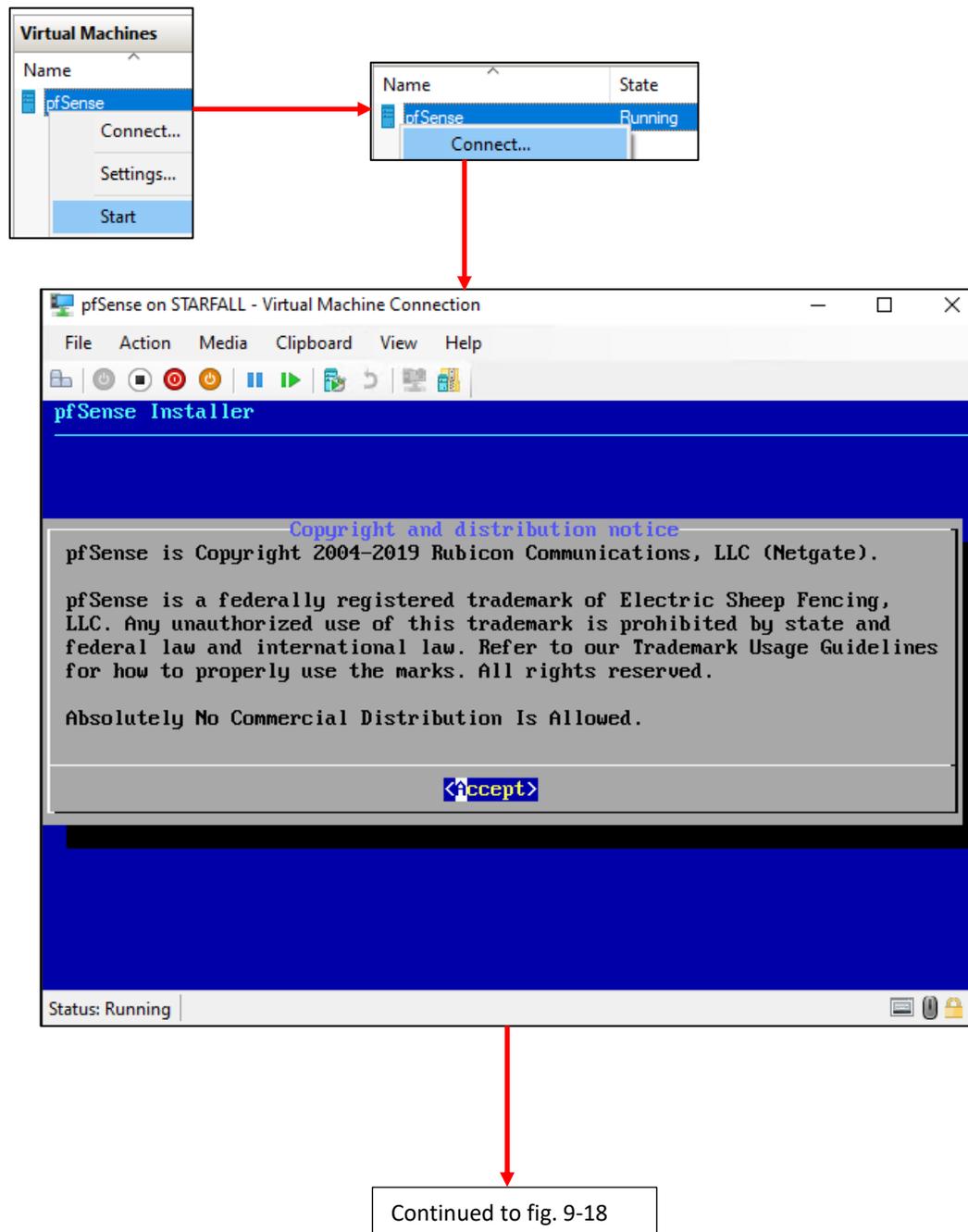
The next step is for students to boot the pfSense VM and install the pfSense firewall distribution to the virtual machine. Right click on pfSense under the *Virtual Machines* section of the *Hyper-V Manager*, and click the *Start* option to boot the virtual machine. Afterwards, right-click on the pfSense virtual machine listing again, and select *Connect* to open up its virtual console. Alternatively, students can right click on the pfSense entry, click *Connect* to open up the virtual console, then click the start button in the virtual console to boot the VM.

Think of the *Virtual Machine Connection* window (e.g., the virtual console) as a direct keyboard, video, and mouse connection to the virtual machine while it is running. You'll notice a lot of text flying by as the VM boots from the installation ISO. Eventually you will reach the pfSense Installer. The first screen shows the *Copyright and distribution notice* for the software. Click anywhere in the virtual console window, and hit Enter to accept the software terms and conditions (without reading them, as is tradition).

Next is the *Welcome* screen for the OS Installer. The option *Install pfSense* should be highlighted by default, but if not, use the arrow keys on your keyboard to select it, then hit enter. The next screen, titled *Keymap Selection* appears. If students are from a region of the world with a unique keyboard layout, they will need to search for and select it. Otherwise, select *Continue with default keymap* to use the US keymap, and hit enter. Next is the *Partitioning* screen. Partitioning is used to tell the installer how much and what portion of the disk to allocate. Since this is a virtual machine, and the disk is relatively tiny (5GB), select *Auto (UFS) Guided Desk Setup* and press enter to tell the installer to use all of the available disk space.

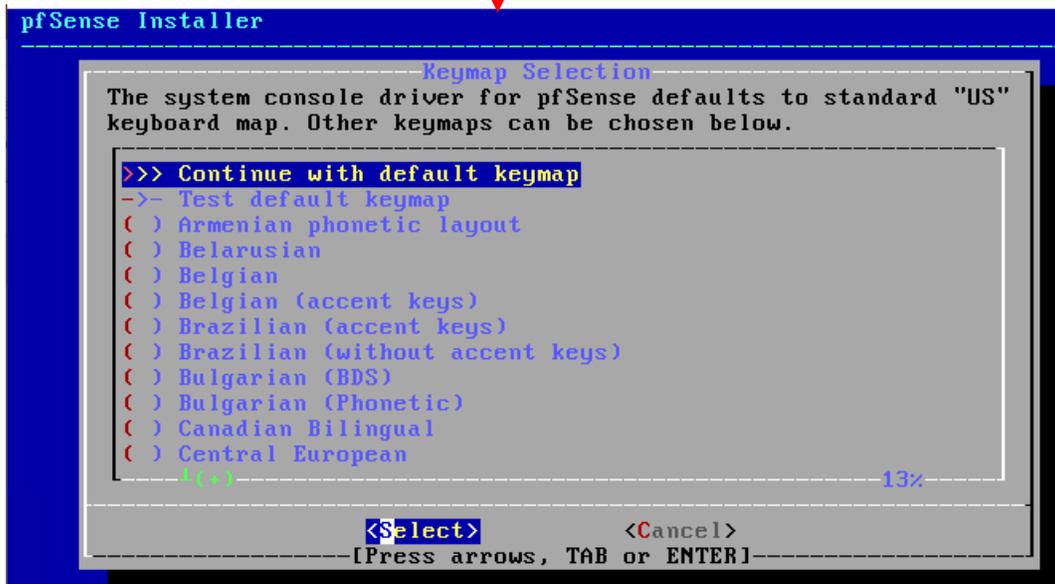
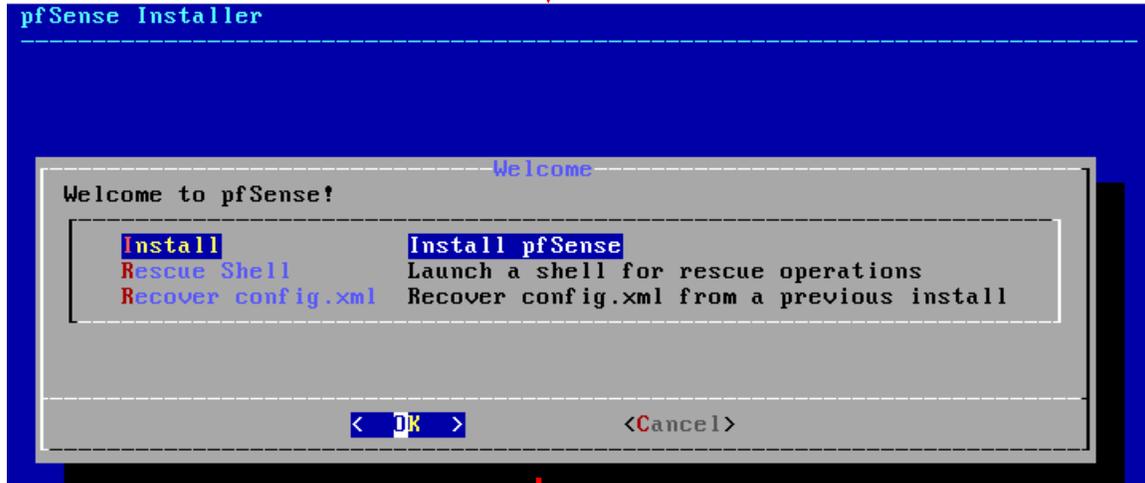
The installer handles formatting the disk and copying the operating system files over. The next screen, titled *Manual Configuration* asks if you want a command shell to manually edit any operating system files before closing the installer. Select *No*, and hit enter again. Finally, on the *Complete* screen, select *Reboot*, and hit enter. Congrats! You just installed the pfSense firewall distribution to your virtual machine.

While the system is rebooting, click on the *Turn Off* option on the virtual console's menu bar (the button is a white circle with black square in the center. When students hover over it, a small pop-up box with the text *Turn Off* appears.). Remember the mouse release key combination is ctrl + alt + left arrow if the virtual console has taken control of the mouse. When students click the *Turn Off* option, a pop-up window labeled *Turn Off Machine* appears asking if the user is sure they want to power off the VM. Click the *Turn Off* button to proceed, and optionally click on the *Please don't ask me again* checkbox to ensure the hypervisor doesn't have the audacity to question your life choices again.



9-17: Right click on the pfSense VM entry in the Hyper-V Manager, and click *Start*. Afterwards, right click again, and select *Connect*. This powers on the pfSense VM, and connects students to the virtual console. The first screen of the pfSense OS installer has a license agreement users must accept to continue. Hit the enter key to accept the license agreement without reading it (as is tradition).

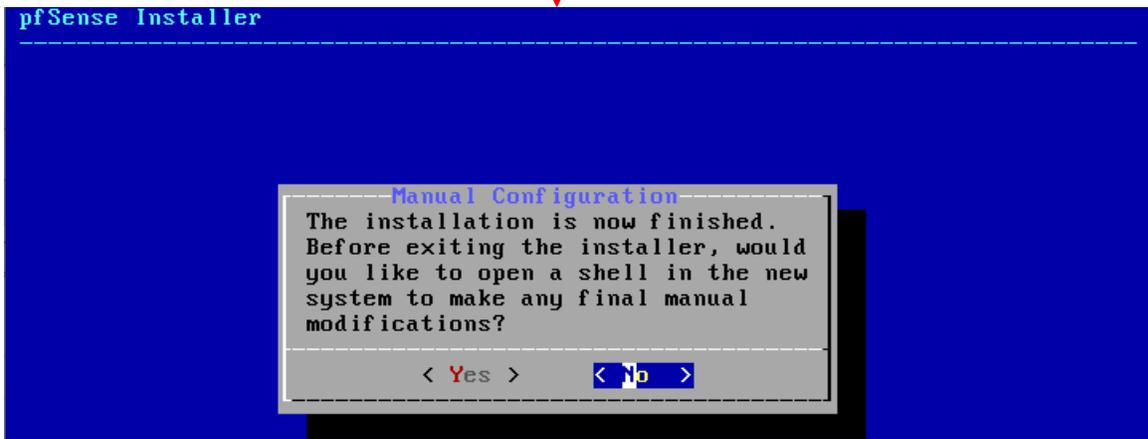
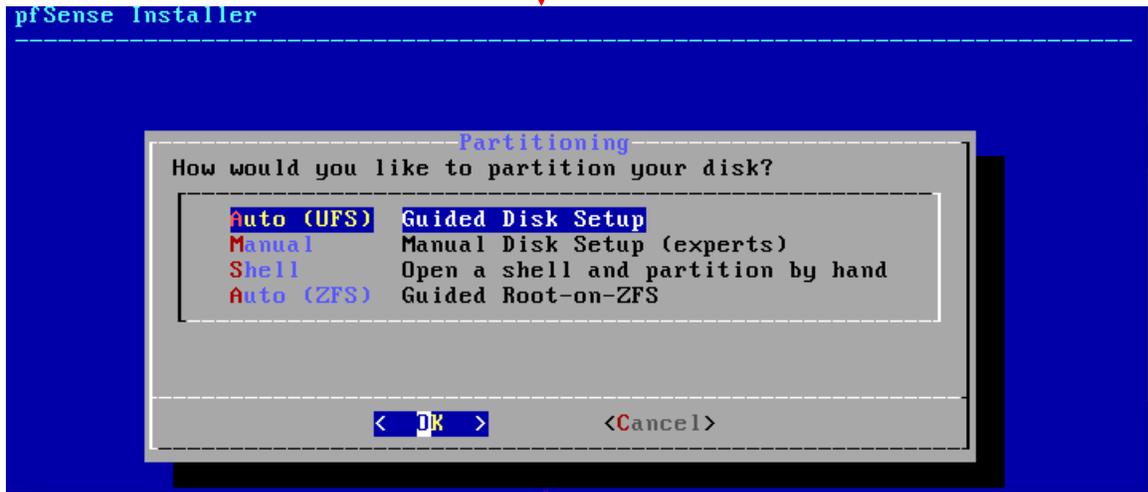
Continued from fig. 9-17



Continued to fig. 9-19

9-18: On the welcome screen, use the arrow keys to highlight *Install*, then hit enter to proceed. Afterwards, students are given the option to select a keymap. Most people can hit enter here to continue with the default US English keymap, but for those who are from different regions can use the arrow keys to find and select a keyboard map, or have pfSense to automatically detect with the *Test default keymap* option.

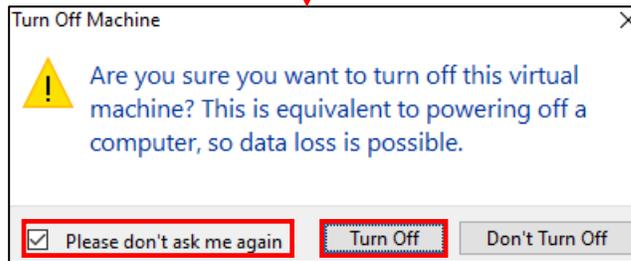
Continued from fig. 9-18



Continued to fig. 9-20

9-19: On the partitioning screen select the *Auto (UFS)* option that is highlighted by default, and pfSense will format and install itself on the virtual hard disk. Afterwards students are asked if they need to perform any manual configuration steps on the newly installed operating system. Select *No* to continue.

Continued from fig. 9-19



9-20: Finally, the operating system installation is complete. Ensure the *Reboot* option is highlighted, and then hit the enter key to begin rebooting the VM. Afterwards, click the icon that looks like a white circle with a black square in the center (when hovered over, a pop-up with the text *Turn Off* will appear) on the virtual console's navigation menu to power off the virtual machine. A pop-up appears asking students if they're sure they want to perform this action. Click the button labeled *Turn Off* to continue powering off the VM. To ensure the hypervisor never dares to question your competence ever again, click the *Please don't ask me again* checkbox.

9.4.4 pfSense Virtual Machine Settings (Part 2)

With the pfSense virtual machine powered off, access its settings menu (in the *Hyper-V Manager*, right click on the pfSense entry in the central pane, under *Virtual Machines*). Under the left navigation pane, locate the object labeled *DVD Drive*. It will be listed under one of the two IDE Controller objects.

Click on the *DVD Drive* entry to highlight it, and bring up its settings pane. Towards the bottom of the pane is button labeled *Remove*. Click this button to remove the *DVD Drive* item from the pfSense VM.

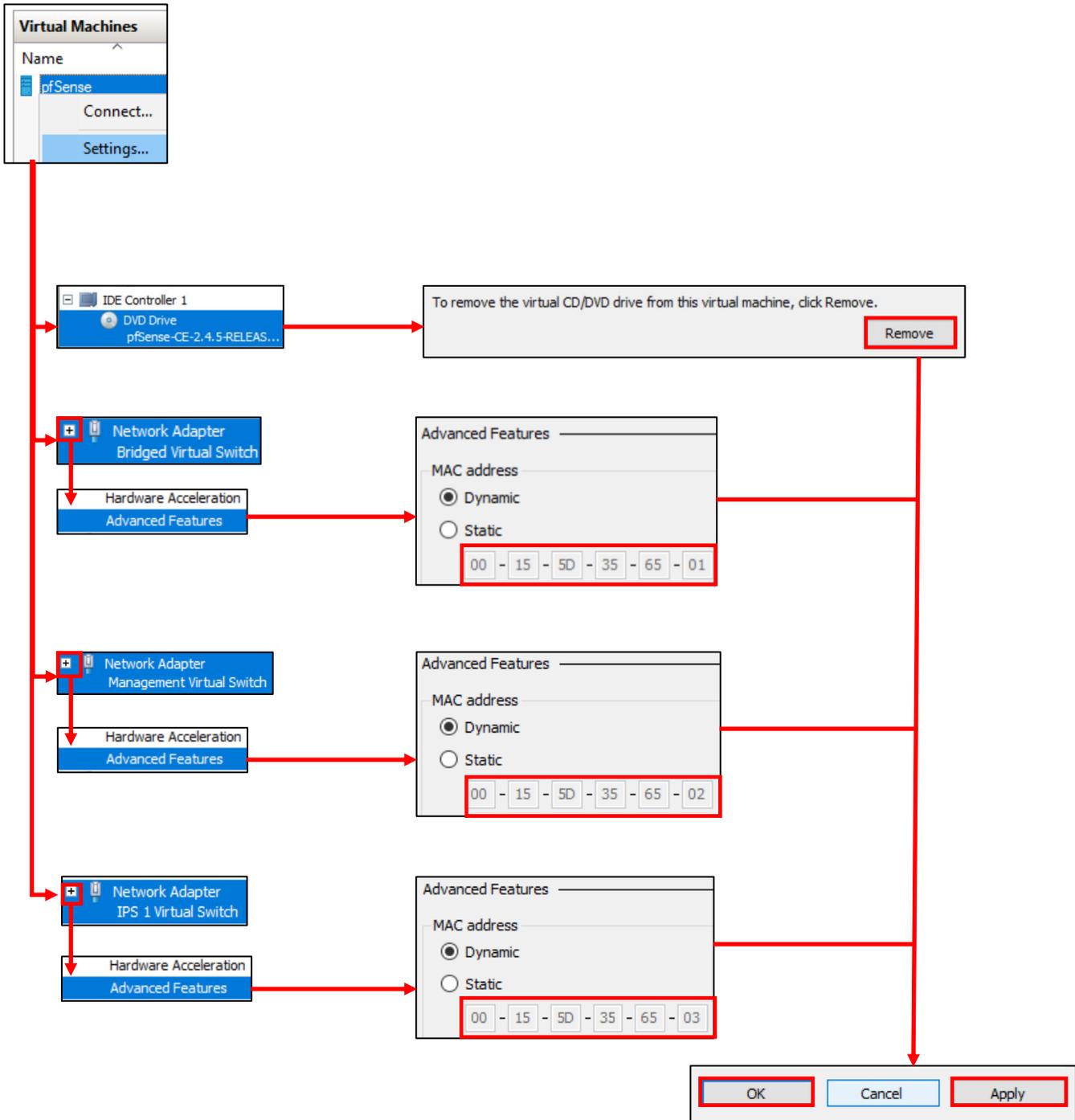
Next, students will need to record the MAC address of each network adapter on the pfSense VM, and note which virtual switch that MAC address is associated with. I will walk students through this process, and use the network adapter attached to the *Bridged Virtual Switch* as an example.

Left click on the *Network Adapter* entry attached to the *Bridged Virtual Switch* to highlight it. Click the small + sign next to it to bring up two sub-entries underneath it, labeled *Hardware Acceleration*, and *Advanced Features*. Click on the *Advanced Features* entry to highlight it, and bring up the configuration options on the right pane. Under the section labeled *MAC address*, there are two radio buttons labeled *Dynamic* and *Static*, and under the *Static* radio button is a series of small input boxes. There should be six in total with two characters in each box. For example, the MAC address of the network adapter attached to the bridged virtual switch for my pfSense VM reads: 00-15-5D-35-65-01. Document the MAC address, note that it belongs to the pfSense VM, and that it is attached to the *Bridged Virtual Switch*. **It is not necessary to change the radio button from Dynamic to Static.**

Students will need to repeat this process two more times for the remaining network adapters. Record the MAC address of each network adapter and document which virtual switch that MAC address is connected to. Once finished, Click *Apply*, then *OK* to leave the *Settings for pfSense* menu. Students should be aware that it is extremely important to document details about the virtual machines connected to their lab environment. Therefore, I highly recommended reading the sidebar conversation, *Noting the Notable* down below.

Backtracing

If the *Advanced Features* menu seems familiar, it's because we were here in section 9.4.2. You're probably wondering: *Why didn't we just record the MAC addresses of all three network interfaces back then?* It's mainly due to laziness on the part of the hypervisor. See, when Hyper-V first creates a network adapter, it doesn't assign it a MAC address until the first time the virtual machine is booted. This means that the MAC addresses you recorded in the section above were not present until after we started the pfSense virtual machine in order to install its operating system. This is will be somewhat important later for the remaining virtual machines in our lab environment, so just keep this in mind as we proceed.



9-21: Open up the *Settings for pfSense* menu again. Locate the *DVD Drive* entry on the left pane, and click the Remove button. Afterwards, locate the network adapters. For each network adapter, click the little "+" sign next to each adapter, then click the *Advanced Features* sub-menu. Under the *MAC address* section, you'll notice two radio buttons and a set of six greyed out input boxes. Record the contents of those input boxes, and note which network adapter and virtual switch that MAC address is associated with. ***It is not necessary to toggle the radio button from Dynamic to Static.*** Once finished, *Apply* the configuration changes, then click *OK* to close the settings menu.

Noting the Notable

I can't overstate the value of documenting your lab network properly. Use whatever note-taking methods you prefer – paper and pen, Evernote, text editors, personal wikis, databases, spreadsheets, etc. Document the name of the VM, Operating system, the number of CPU cores allocated, RAM, Disk size, number of network adapters, network segments they are attached to, and their MAC addresses. This is called *asset management*, and it's an important habit to cultivate. Here is a template you can use for documenting your VMs:

VM Name:
Operating System:
CPU Cores:
RAM:
Disk Size:
Virtual Network Adapters:
Network Adapter #:
-Network Segment:
-MAC Address:
<Repeat for each network adapter>
Additional Notes:

And as an example, here is my pfSense VM entry:

VM Name: pfSense
Operating System: pfSense (FreeBSD)
CPU Cores: 1
RAM: 512MB
Disk Size: 5GB
Virtual Network Adapters: 3
Network Adapter 1:
-Network Segment: Bridged Virtual Switch/WAN
-MAC Address: 00:15:5D:35:65:01
Network Adapter 2:
-Network Segment: Management Virtual Switch/LAN
-MAC Address: 00:15:5D:35:65:02
Network Adapter 3:
-Network Segment: IPS 1 Virtual Switch/OPT1
MAC Address: 00:15:5D:35:65:03
Additional Notes: Lab firewall. Provides NTP, DNS, DHCP,
and HTTP proxy services.

Do this for every single virtual machine you add to your lab environment. Keep track of systems added or removed from the lab network. Always be aware of what's running on your networks. If you can do these things, you'll be better at asset management than most of the Fortune 500.

9.4.5 pfSense Command-Line and initial interface configuration

In this section, readers will navigate the command-line interface of their pfSense virtual machine to perform essential setup tasks. Once completed, users can navigate to the webConfigurator interface. Start the pfSense VM, then connect to its virtual console (refer to [section 9.4.3](#), pp. 112-116 if students need a refresher).

9.4.5.1 The Assign Interfaces Wizard

After a few moments, the boot process completes and students are greeted by the *Assign Interfaces wizard*.

Note: If by some chance the *Assign Interfaces* wizard didn't start, or it otherwise exited, the pfSense command-line menu should be displayed. Select option number 1 to start the wizard manually.

This wizard is used to map our virtual machine's network interfaces (*Adapter 1*, *Adapter 2*, and *Adapter 3*) to their pfSense aliases – *WAN*, *LAN*, or *OPT1*. Unfortunately, the operating system itself also has unique names for each of these interfaces, adding another layer of complexity and confusion when trying to perform this task.

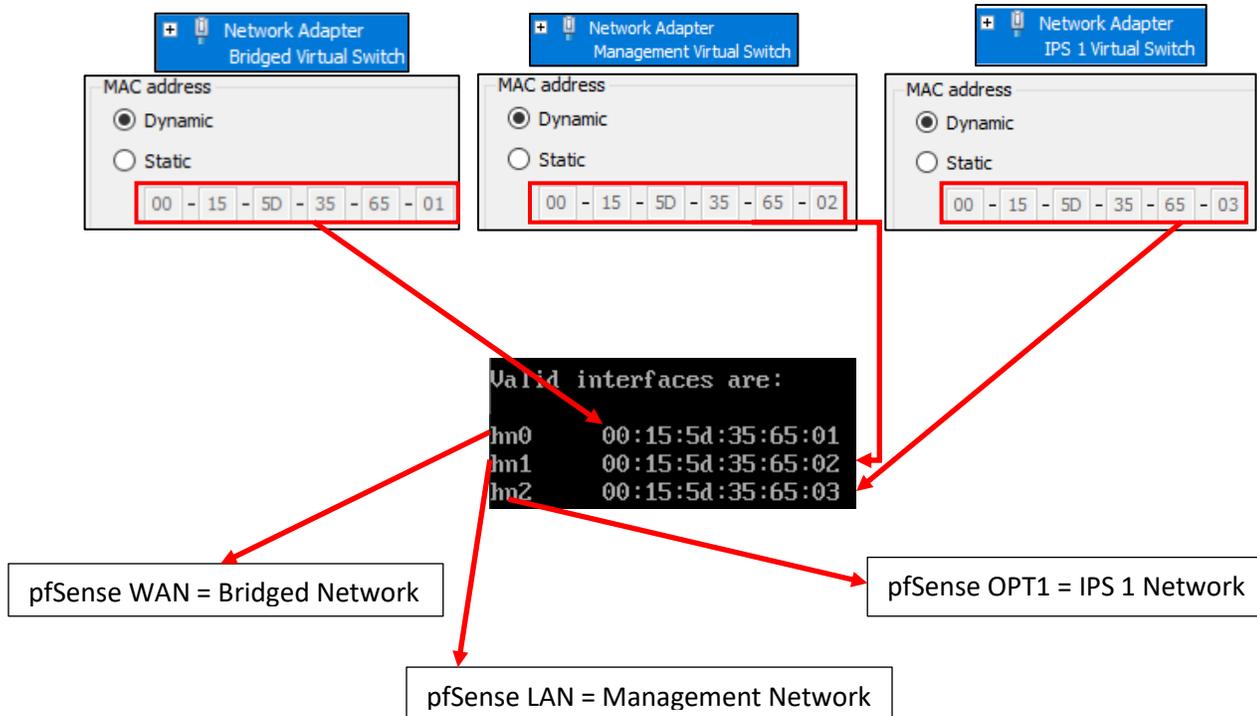
pfSense itself is based on the FreeBSD operating system, and BSD has its own methods for assigning physical (or virtual, in our case) network interfaces an interface name. For example, BSD assigned the network adapters of my virtual machine the interface names *hn0*, *hn1*, and *hn2*. Every network adapter – integrated or not, virtual or physical, wired or wireless – all have a MAC address to uniquely identify them on a local network. We're going to take advantage of that to know for certain which of the three interfaces, *hn0* through *hn2*, map to *network adapters 1* through *3*, and how they should be assigned as the *WAN*, *LAN* and *OPT1* aliases. Students were highly advised to record the contents of the *MAC Address* input boxes of all three network adapters to assist in this task.

A quick way for readers to determine the interface names for their pfSense installation is through the wizard itself. A section of text labeled *Valid interfaces are* appears, followed by a series of lines. These lines provide the interface names, MAC addresses, current operational status, and type of hardware BSD identifies the network interface as (The drivers BSD loaded) for each network interface pfSense was able to detect. ***Students should have 3 of these lines in total.*** Here is some sample output from my lab environment:

```
Valid interfaces are: 2
hn0 1 00:15:5d:35:65:01 (down) Hyper-V Network Interface
hn1 00:15:5d:35:65:02 (down) Hyper-V Network Interface
hn2 00:15:5d:35:65:03 (down) Hyper-V Network Interface
```

9-22: A portion of the *Assign Interfaces* wizard. Pay attention to the interface names (1) and the MAC addresses for those interface names (2). This information is needed to determine which virtual network segment they are connected to. This in turn allows students to assign the *WAN*, *LAN* and *OPT1* interfaces correctly.

Compare the MAC addresses displayed, to the MAC addresses recorded earlier, and use that information to complete the rest of the wizard. A diagram (fig. 9-23) is provided below to help students understand how to correctly perform this mapping process.



9-23: Here we have the network configuration for my pfSense VM, and the output from the valid interfaces table from the *Assign Interfaces* wizard. The bridged virtual switch adapter has the MAC Address 00-15-5D-35-65-01. Looking at the valid interfaces table, hn0 has the same MAC address, just with colons (:) every 2 characters (the correct notation for MAC addresses). hn0 should be assigned as the WAN interface. The MAC address of the adapter attached to the management virtual switch matches the MAC address for hn1. This means hn1 should be assigned the LAN interface. Finally, the adapter connected to the IPS 1 virtual switch matches the MAC address for hn2. This means that hn2 should be assigned the OPT1 interface.

The remainder of this section will aim to guide students through the various questions the wizard will ask (in *italicized* font), and the answers I provided (in *bold* font) based on my lab network and adapter to MAC address mappings. **Students should be aware that this is by and far the most important configuration task for pfSense.** Making sure that the Hyper-V network adapters map to the correct pfSense aliases and network segments is absolutely vital to the lab environment working correctly.

Should VLANs be set up now [y|n]? n

*Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): **hn0***

*Enter the LAN interface name or 'a' for autodetection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): **hn1***

*Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): **hn2***

The interfaces will be assigned as follows:

*WAN -> **hn0***

*LAN -> **hn1***

*OPT1 -> **hn2***

*Do you want to proceed [y|n]? **y***

After answering these questions, pfSense will bring students to the command-line menu. This interface consists of a series of options, numbered one through sixteen that users can access by inputting the number of the option they desire.

```

Network interface mismatch -- Running interface assignment option.

Valid interfaces are:

hn0      00:15:5d:35:65:01 (down) Hyper-V Network Interface
hn1      00:15:5d:35:65:02 (down) Hyper-V Network Interface
hn2      00:15:5d:35:65:03 (down) Hyper-V Network Interface

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y/n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(hn0 hn1 hn2 or a): hn0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(hn1 hn2 a or nothing if finished): hn1

Enter the Optional 1 interface name or 'a' for auto-detection
(hn2 a or nothing if finished): hn2

The interfaces will be assigned as follows:

WAN  -> hn0
LAN  -> hn1
OPT1 -> hn2

Do you want to proceed [y/n]? y
FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

Hyper-V Virtual Machine - Netgate Device ID: 5f3170459a9890f0468a

*** Welcome to pfSense 2.4.5-RELEASE (amd64) on pfSense ***

WAN (wan)      -> hn0      -> v4/DHCP4: 10.0.0.19/24
                v6/DHCP6: 2601:408:502:c330:215:5dff:fe35:6501
/64
LAN (lan)      -> hn1      -> v4: 192.168.1.1/24
OPT1 (opt1)    -> hn2      ->

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option:

```

9-24: A selection of screen captures from the *Assign Interfaces* wizard, stitched together to show the questions the wizard asks, and the responses based on network adapter mappings in *fig. 9-23*. This is what students see upon first booting into pfSense. When finished, students are greeted with the pfSense command-line menu.

9.4.5.2 Setting IP Addresses for WAN, LAN, and OPT1

The next task we will need to perform on the pfSense command-line is assigning IP addresses to the *WAN*, *LAN*, and *OPT1* interfaces using the *Set interface(s) IP address* wizard. Most students will have their host system connected to a home or enterprise network where DHCP is available, and just about anything that requests an IP address lease will get one with no problems. That *should* include the pfSense *WAN* interface bridged to that network. This means the *WAN* interface should already have an IP address, subnet mask, default gateway (and usually, DNS servers to forward DNS requests to) automatically provided (if this is not the case, see the sidebar discussion, *Help! The WAN Interface has no IP Address*, for some troubleshooting pointers). That means we should only have to run through the *Set interface(s) IP address* wizard twice – once for the *LAN* interface, and once for the *OPT1* interface. Select option 2 from the pfSense menu to get started.

Similar to the previous section, the remainder of this section is going to consist of the questions the *Set interface(s) IP address* wizard will ask students (*italicized*), and the correct answers for the *LAN* and *OPT1* interfaces (in **bold**), followed by an illustration depicting the same questions and answers.

LAN interface:

Available interfaces:

- 1 – WAN (*[interface name] – [dhcp/dhcp6/static address configuration]*)
- 2 – LAN (*[interface name] – static*)
- 3 – OPT1 (*[interface name]*)

Enter the number of the interface you wish to configure: **2**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.1.1**

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense

e.g. 255.255.255.0 = 24

255.255.0.0 = 16

255.0.0.0 = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.1.10**
Enter the end address of the IPv4 client address range: **172.16.1.254**
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24

You can now access the webConfigurator by opening the following URL in your web browser:

<https://172.16.1.1>

Press <ENTER> to continue. <ENTER>

```
Enter the number of the interface you wish to configure: 2
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 172.16.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
    255.255.0.0   = 16
    255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.1.10
Enter the end address of the IPv4 client address range: 172.16.1.254
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
    https://172.16.1.1/

Press <ENTER> to continue.
```

9-25: Screen captures from the *Set interface(s) IP address wizard*, stitched together to show the questions the wizard asks, and the responses for the LAN interface based on network adapter mappings in *fig. 9-23*.

Here is an abridged set of *questions* and **answers** for the *OPT1* interface:

OPT1 interface (abridged):

Available interfaces:

- 1 – WAN ([interface name] – [dhcp/dhcp6/static address configuration])
- 2 – LAN ([interface name] – static)
- 3 – OPT1 ([interface name])

Enter the number of the interface you wish to configure: **3**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.2.1**

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

*Do you want to enable the DHCP server on LAN? (y/n) **y***

*Enter the start address of the IPv4 client address range: **172.16.2.10***

*Enter the end address of the IPv4 client address range: **172.16.2.254***

*Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n***

```

Enter the number of the interface you wish to configure: 3
Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 172.16.2.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new OPT1 IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new OPT1 IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on OPT1? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.2.10
Enter the end address of the IPv4 client address range: 172.16.2.254

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to OPT1...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 OPT1 address has been set to 172.16.2.1/24
Press <ENTER> to continue.

```

9-26: Screen captures from the *Set interface(s) IP address* wizard, stitched together to show the questions the wizard asks, and the responses for the *OPT1* interface based on network adapter mappings in *fig. 9-23*.

After running the wizard again for the *OPT1* interface, students should have an IP address for the *WAN*, *LAN* and *OPT1* interfaces. Additionally, DHCP ranges should be assigned for the *LAN* and *OPT1* interfaces. We're just about ready to move to the webConfigurator, but before doing so, let's run some network connectivity tests first.

```

WAN (wan)      -> hn0      -> v4/DHCP4: 10.0.0.19/24
                v6/DHCP6: 2601:408:502:c330:215:5dff:fe35:6501
/64
LAN (lan)      -> hn1      -> v4: 172.16.1.1/24
OPT1 (opt1)    -> hn2      -> v4: 172.16.2.1/24

```

9-27: The interface information portion of the pfSense command-line menu should look something like this. Looking good is one thing, now let's see if it actually works.

What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?

Unfortunately, I have no way of knowing what network ranges students use at home, so it's entirely possible your physical network may already be using one of the ranges I'm asking you to configure for your lab environment (e.g., 172.16.1.0/24, or 172.16.2.0/24). **To avoid network conflicts on your home network, try these alternate configurations for the *Set interface(s) IP address wizard*:**

Alternate LAN configuration:

LAN interface IP address: 172.16.11.1
Subnet mask bit count: 24
DHCP start address: 172.16.11.10
DHCP end address: 172.16.11.254

Alternate OPT1 configuration:

OPT1 interface IP address: 172.16.12.1
Subnet mask bit count: 24
DHCP start address: 172.16.12.10
DHCP end address: 172.16.12.254

If your lab network is connected to a school or enterprise network using the entire 172.16.0.0/12 allocation, things may be a little more complicated. It may be best to use one of the other RFC1918 network allocations instead, such as 192.168.0.0/16, or 10.0.0.0/8. Why? Enterprise networking can become complicated, either due to growth over time, legacy configurations, or work-arounds to problems accrued over time. You don't want to troubleshoot network problems on your host system, nor do you want the IT ops team coming to your desk over a network outage that could've been avoided. **Here are some alternate configurations for the *Set interface(s) IP address wizard* if you need to avoid using 172.16.0.0/12 entirely:**

Alternate LAN configuration 1:

LAN interface IP address: 10.0.11.1
Subnet mask bit count: 24
DHCP start address: 10.0.11.10
DHCP end address: 10.0.11.254

Alternate OPT1 configuration 1:

LAN interface IP address: 10.0.12.1
Subnet mask bit count: 24
DHCP start address: 10.0.12.10
DHCP end address: 10.0.12.254

Alternate LAN configuration 2:

LAN interface IP address: 192.168.11.1
Subnet mask bit count: 24
DHCP start address: 192.168.11.10
DHCP end address: 192.168.11.254

Alternate OPT1 configuration 2:

LAN interface IP address: 192.168.12.1
Subnet mask bit count: 24
DHCP start address: 192.168.12.10
DHCP end address: 192.168.12.254

Substituting Instructions for Your Chosen Network Ranges

Keep in mind you don't have to use the alternate configurations recommended above. If students have some experience with networking and subnetting, they're welcome to use any network range that suits them. These are just some suggestions to help those who are not quite as experienced, and want to avoid network conflicts.

As a final reminder, **the remaining sections, chapters, and configuration steps will all assume that readers are using 172.16.1.0/24 for the LAN network and 172.16.2.0/24 for the OPT1 network.** This means you will have to mentally substitute steps and commands for the network range you are using instead.

For example, the lab network diagram in chapter 6 has the Kali VM on the IPS 1 (OPT1) network, with an IP address of 172.16.2.2. If you are using an alternate network configuration for the OPT1 network, say 192.168.12.0/24, then the Kali VM's IP address should be 192.168.12.2. If I say "*run the command ssh username@172.16.2.2 to connect to the kali VM*", you'll have to mentally substitute that with `ssh username@192.168.12.2` instead. As another example, firewall rules denying access to or from 172.16.2.3 (Metasploitable 2) should be created for 192.168.12.3 instead. Keep this in mind as you continue to build your lab network!

Help! The WAN Interface has no IP Address

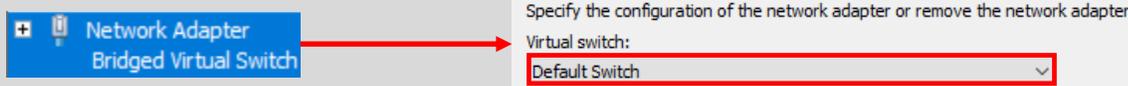
If the WAN interface of your pfSense VM has no IP address, consider some of the following to help with troubleshooting:

-No DHCP – It's pretty rare, but perhaps the WAN interface is bridged to a network without DHCP. This just means that you'll have to run the *Set interface(s) IP address* wizard to manually configure the WAN interface IP address, subnet mask, and default gateway. I've already listed the questions the wizard asks, and provided the answers for the LAN and OPT1 interfaces, but since I have absolutely no idea what IP address and subnet mask is assigned to your local physical network, I cannot tell you what you need to enter for the wizard.

If you don't know either, ask a network administrator or whoever is responsible for your network to assist you. Note that if required to manually configure these settings here, practically all of the tasks that require DNS to be configured (e.g., network connectivity tests, and checking for updates on your VMs) will not work until DNS server addresses are configured. This can be done via the webConfigurator, and will be covered shortly.

-Incorrect Virtual Switch Configuration – Another possibility is perhaps the Bridged Virtual Switch that the WAN network adapter is connected to may be misconfigured. Refer to [section 9.3.2](#) (pp. 96-99) to access the *Virtual Switch Manager*. Confirmed the *Bridged Virtual Switch* type is set to *External*, and that it is bridged to the correct physical network interface on the host operating system. Additionally, make absolutely sure the pfSense WAN interface was mapped to the correct network virtual switch. Refer to [section 9.4.5.1](#) (pp. 120-123) for further guidance.

-NAC Interference – If you're network security enthusiast at home or connected to an enterprise network, NAC (network access control) may be preventing the WAN interface from obtaining an IP address. Back in chapter 4, [section 4.1.2, NAT Networking \(and Port Forwarding\)](#) (pp. 46-47), readers learned how network address translation works, and how in situations like this, you may be forced to use NAT network options to work around network security. If you suspect the WAN interface is being blocked, you can try editing the pfSense VM's settings for the adapter attached to the bridged virtual switch, and instead attach it to the default switch. If you choose to this, Reboot the pfSense VM. It should have an IP address from the NAT network, but that doesn't mean it has network connectivity. Continue below to the next section as normal to test your network connectivity.



Note: If this doesn't work, or attempting to subvert network access controls would otherwise get students in trouble, consider talking to your network/systems administrator and seeing if you can get DHCP allocation and/or necessary exceptions put into place. **Don't violate acceptable use policies, and don't break the law.**

9.4.6 Testing Internet Connectivity using Shell commands

Select option 8, labeled *Shell* in the pfSense menu. Doing so will open up a command-line (bash) shell. Run these 3 commands, and observe their output:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

Here is output from these 3 commands:

```
Enter an option: 8

[2.4.5-RELEASE][root@pfSense.localdomain]/root: ping -c 4 www.google.com
PING www.google.com (172.217.6.100): 56 data bytes
64 bytes from 172.217.6.100: icmp_seq=0 ttl=54 time=24.496 ms
64 bytes from 172.217.6.100: icmp_seq=1 ttl=54 time=22.714 ms
64 bytes from 172.217.6.100: icmp_seq=2 ttl=54 time=21.638 ms
64 bytes from 172.217.6.100: icmp_seq=3 ttl=54 time=19.490 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 19.490/22.084/24.496/1.813 ms
[2.4.5-RELEASE][root@pfSense.localdomain]/root: nslookup www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.6.100
Name:   www.google.com
Address: 2607:f8b0:4009:812::2004

[2.4.5-RELEASE][root@pfSense.localdomain]/root: curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Mon, 01 Jun 2020 02:53:41 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Mon, 01 Jun 2020 02:53:41 GMT
cache-control: private
set-cookie: 1P_JAR=2020-06-01-02; expires=Wed, 01-Jul-2020 02:53:41 GMT; path=/;
domain=.google.com; Secure
set-cookie: NID=204=DdNU16afHrYu25Utm83temwvvrSe6a4UyA3YHz_JKLFzBAv7xrWi8HjSn2-x1
PNmxh3EutjAoFBh15hNpxrU72.jpzLLQU0JHJxaOMh5mFyntk5Gae7KUMe2-d1g8I1KloIb7HzOBP_BB4
b0sb4lt0Tv1zwOdriUE8ndqfygcrN04; expires=Tue, 01-Dec-2020 02:53:41 GMT; path=/;
domain=.google.com; HttpOnly
alt-svc: h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=25
92000,h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma=2
592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=259
2000; v="46,43"
```

9-28: The output from the `ping -c 4`, `nslookup`, and `curl -I` commands. All three of these commands completed successfully. Pay close attention to the marked sections above. Note that the IP addresses returned for `nslookup` (In the fields labeled *Address*) may vary based on region.

In a nutshell, these three commands are being used to test various forms of internet connectivity for our VM. `ping -c 4 www.google.com` tells pfSense to send 4 (and only 4) ICMP packets to a specific destination, requesting that the destination respond with its own ICMP packets if it has been reached. `nslookup www.google.com` asks our pfSense virtual machine's configured DNS servers to translate a domain name to an IP address for us. Finally, `curl -I https://www.google.com` is being used to test HTTPS connectivity to the internet. The `-I` option tells the command to only return the HTTP Server headers from our request. All we're really interested in is the line of text: `HTTP/2 200`. This is a thumbs up from Google's webserver confirming that they got our HTTPS request with no problems.

Students already familiar with DNS basics may have noticed that we are already trying to ping a domain name (`www.google.com`) with our `ping` command. This means, that in order to actually ping the correct destination, our virtual machine will need to make a DNS request to find the IP address of `www.google.com`. That makes the `nslookup` test redundant, right? Well, yes and no. Later in this chapter, as new virtual machines get created, readers will be advised to perform connectivity tests on those VMs as well. However, the pfSense firewall policy is going to be very strict, so ICMP packets outbound from our lab network will be blocked. Due to how DNS works, the `nslookup` check can still be used to make sure VMs can resolve domain names, and the `curl` connectivity test will be more than sufficient to confirm whether or not lab virtual machines have the internet access they require. After performing these commands and confirming internet connectivity, type `exit` to leave the shell.

My connectivity commands failed! Now what?

If students got anything other than output similar to *fig. 9-28* (e.g., request timeouts and/or packet loss for `ping`, timeouts for `nslookup`, no response for `curl -I`), then there are connectivity issues to be sure. Troubleshooting network connectivity is an extremely complex topic. I can't give you a definitive guide for finding the root of your problem, but I can tell you to start with the basics and work your way up – sometimes the cause of your network problems are settings or hardware that was taken for granted.

Begin by checking physical cabling, link lights, and physical connectivity to network devices first. As an extension to that, check out the sidebar in section 9.4.5.2 (*Help! The WAN interface has no IP address*) for some additional clues. The *Bridged Virtual Switch* may be bridged to the wrong physical adapter. Some form of network security (e.g., a network firewall) may be preventing your VM from connecting to the internet. Try connecting the *Bridged Virtual Switch* network adapter to the default (NAT) virtual switch instead. The incorrect network adapter may have been chosen to be the WAN interface. Consider re-running the *Assign Interfaces* wizard again, and compare the MAC addresses from the wizard to the MAC addresses of the network adapter in the pfSense VM's Settings menu. See *fig. 9-23* for guidance on confirming that interfaces have been mapped correctly.

If students were required to run the *Set interface(s) IP address* wizard for the WAN interface (No DHCP), or your local network's DHCP server doesn't assign DNS servers automatically, your troubleshooting commands will fail because pfSense has no way of resolving domain names. We will be covering how to manually configure a primary and/or secondary DNS server for pfSense via the webConfigurator shortly.

If your host system is connected to a physical network already using 172.16.1.0/24 or 172.16.2.0/24, you may be experiencing network conflicts, routing loops, or other weird behavior. Assign different IP addresses and network ranges to the LAN and OPT1 networks to avoid network conflicts. See the sidebar discussion in 9.4.5.2 labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?*

Last but not least, check and double check that you entered the commands correctly. Typos matter on the command-line, and BSD will not hold your hand if the command is entered incorrectly. If all else fails, don't be afraid to ask others for guidance.

9.4.7 Finish setting up pfSense

Navigate to chapter 14, *pfSense Firewall Policy and Network Services*, starting on p. 664 and follow the chapter guidance. Once completed, readers will be directed back here to complete their lab environment.

9.5 Create the Remaining Virtual Machines

Welcome back! Now that the pfSense VM is fully functional, it's time to start working on the remaining lab VMs. In this section, users will create three of the four remaining virtual machines via the *New Virtual Machine* wizard, then adjust the *Settings* of each virtual machine. After the SIEM, IPS and Kali VMs are created and configured, readers will be guided through the operating system installation, and initial setup process for all three VMs. The Metasploitable 2 VM is a unique case, and will be covered separately.

9.5.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali

Run the *New Virtual Machine* wizard three times, with the settings listed below. Assume the default for any settings not mentioned in the table below. Refer back to [section 9.4.1](#) (pp. 102-108) for guidance on how to access and progress through the wizard as needed.

Name:	SIEM	IPS	Kali
Location:	Click the <i>Store the virtual machine in a different location</i> checkbox	Click the <i>Store the virtual machine in a different location</i> checkbox	Click the <i>Store the virtual machine in a different location</i> checkbox
	<Virtual Machines Directory>\SIEM	<Virtual Machines Directory>\IPS	<Virtual Machines Directory>\Kali
Generation:	Generation 1	Generation 1	Generation 1
Memory:	Uncheck <i>Use Dynamic Memory</i>	Uncheck <i>Use Dynamic Memory</i>	Uncheck <i>Use Dynamic Memory</i>
	4GB (4096MB)	4GB (4096MB)	4GB (4096MB)
Networking:	Management Virtual Switch	Management Virtual Switch	IPS 1 Virtual Switch
Virtual Hard Disk Size:	80GB	80GB	80GB
Installation Options:	Select the <i>Install an operating system from a bootable CD/DVD-ROM</i> radio button.	Select the <i>Install an operating system from a bootable CD/DVD-ROM</i> radio button.	Select the <i>Install an operating system from a bootable CD/DVD-ROM</i> radio button.
	Select the <i>Image file (.iso)</i> radio button.	Select the <i>Image file (.iso)</i> radio button.	Select the <i>Image file (.iso)</i> radio button.
	Locate the Ubuntu Server ISO	Locate the Ubuntu Server ISO	Locate the Kali Linux ISO

Note: The *Location* setting has a placeholder, <Virtual Machines Directory>. This refers to the directory students configured for the *Virtual Machines* setting in the *Hyper-V Settings* menu. Check out [section 9.3.1](#) (pp. 92-95) for details. This means if your Virtual Machines directory setting is c:\VMs, that the directory for the SIEM VM should be C:\VMs\SIEM.

Name:	SIEM
Generation:	Generation 1
Memory:	4096 MB
Network:	Management Virtual Switch
Hard Disk:	E:\VM-Configs\SIEM\SIEM\Virtual Hard Disks\SIEM.vhdx (VHDX, dynamically expanding)
Operating System:	Will be installed from D:\Tony\Desktop\lab_isos\ubuntu-20.04.1-live-server-amd64.iso
Name:	IPS
Generation:	Generation 1
Memory:	4096 MB
Network:	Management Virtual Switch
Hard Disk:	E:\VM-Configs\IPS\IPS\Virtual Hard Disks\IPS.vhdx (VHDX, dynamically expanding)
Operating System:	Will be installed from D:\Tony\Desktop\lab_isos\ubuntu-20.04.1-live-server-amd64.iso
Name:	Kali
Generation:	Generation 1
Memory:	4096 MB
Network:	IPS 1 Virtual Switch
Hard Disk:	E:\VM-Configs\Kali\Kali\Virtual Hard Disks\Kali.vhdx (VHDX, dynamically expanding)
Operating System:	Will be installed from D:\Tony\Desktop\lab_isos\kali-linux-2020.2-installer-amd64.iso

9-29: The summary page for the SIEM, IPS, and Kali virtual machines. Please note that the *Use Dynamic Memory* checkbox should be disabled for all three virtual machines. Additionally, the recommended *Virtual Hard Disk Size* for all three VMs is 80GB. Neither of these configurations will be reflected on the summary screen, so students should make absolutely sure that they have configured them properly.

With all three VMs created, students will need to enter the settings menu for all three virtual machines, and make some adjustments:

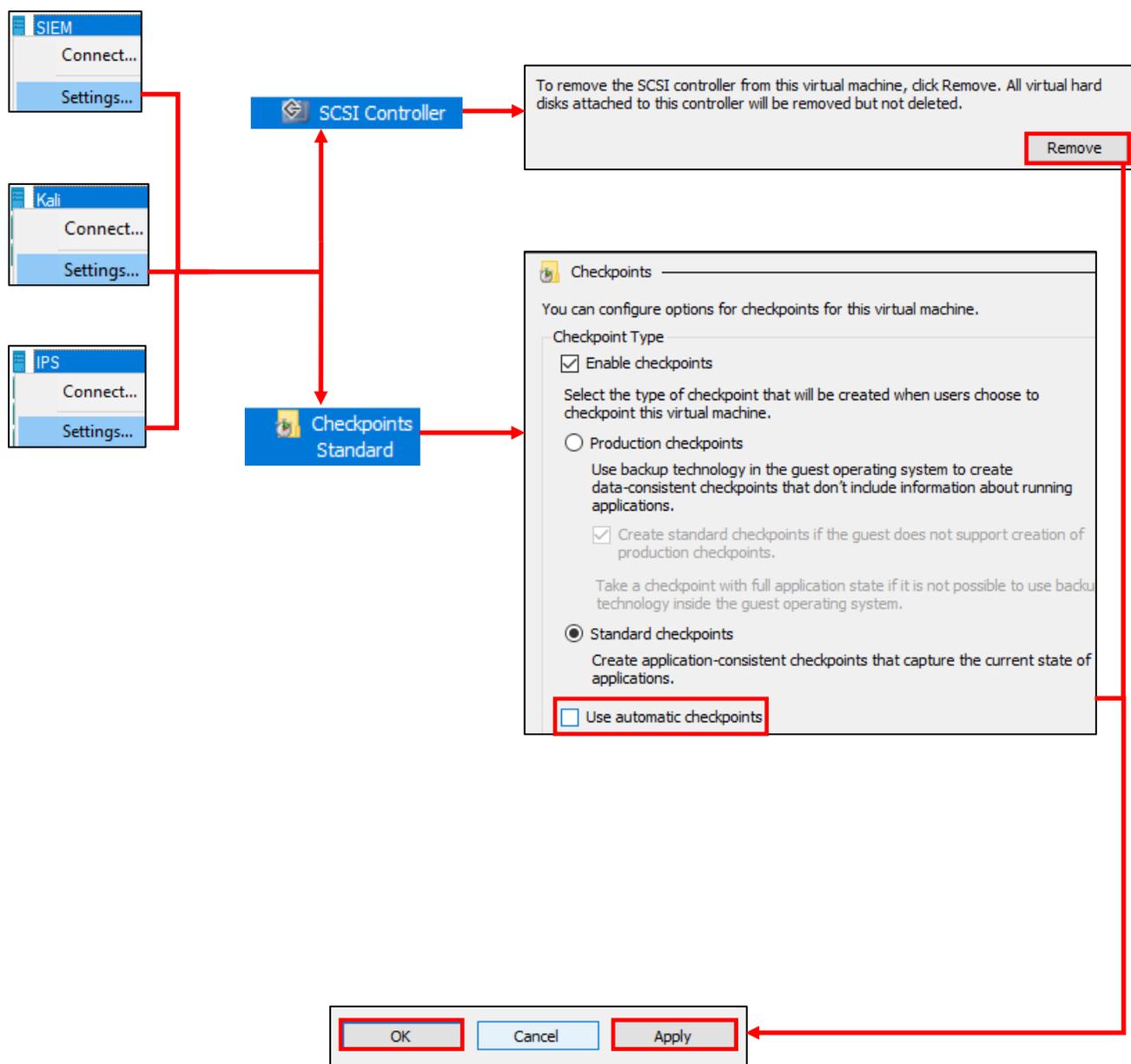
- Remove the *SCSI Controller* virtual hardware
- Under the checkpoints option, ensure the *Enable checkpoints* option is checked, The *Standard checkpoints* radio button is selected, and uncheck the *Use automatic checkpoints* option.

In addition to performing these configuration changes to all three virtual machines, the IPS virtual machine needs two more virtual network adapters. Use the *Add Hardware* option to create to more network adapters with the following settings:

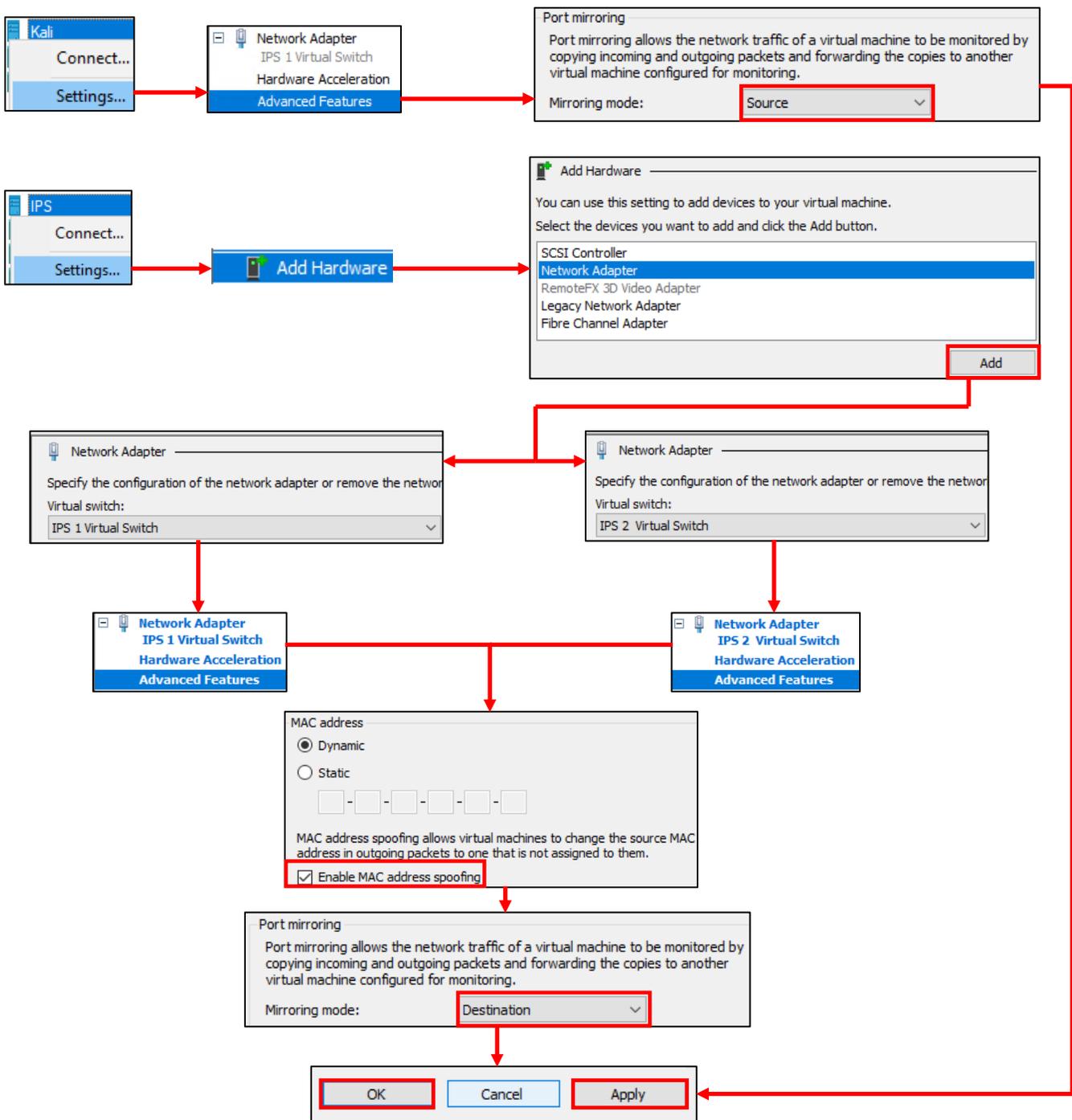
- Attach one of the network adapters to the *IPS 1 Virtual Switch*, and the other to the *IPS 2 Virtual Switch*.
- Under *Advanced Features* for these two new network adapters, check the *Enable MAC address spoofing* checkbox, under the *MAC address* section.
- Under *Advanced Features*, locate the *Port Mirroring* section, and set the *Mirroring mode* drop-down for both adapters to *Destination*.

Finally, access the *Advanced Features* menu of the Kali VM, locate the *Port Mirroring* section, and set its *Mirroring mode* drop-down to *Source*.

The additional network adapters for the IPS VM, and the Advanced Features settings for both the IPS and Kali VMs are absolutely essential. If students need a refresher on how to adjust virtual machine settings, see [section 9.4.2](#) (pp. 109-111) for further guidance.



9-30: Remove the *SCSI Controller* from all three virtual machines, then navigate to the *Checkpoints* option, ensure the *Enable checkpoints* box is selected, that the *Standard checkpoints* radio button is selected, and that the *Use automatic checkpoints* checkbox is unchecked. *Apply* these changes, then click *OK*.



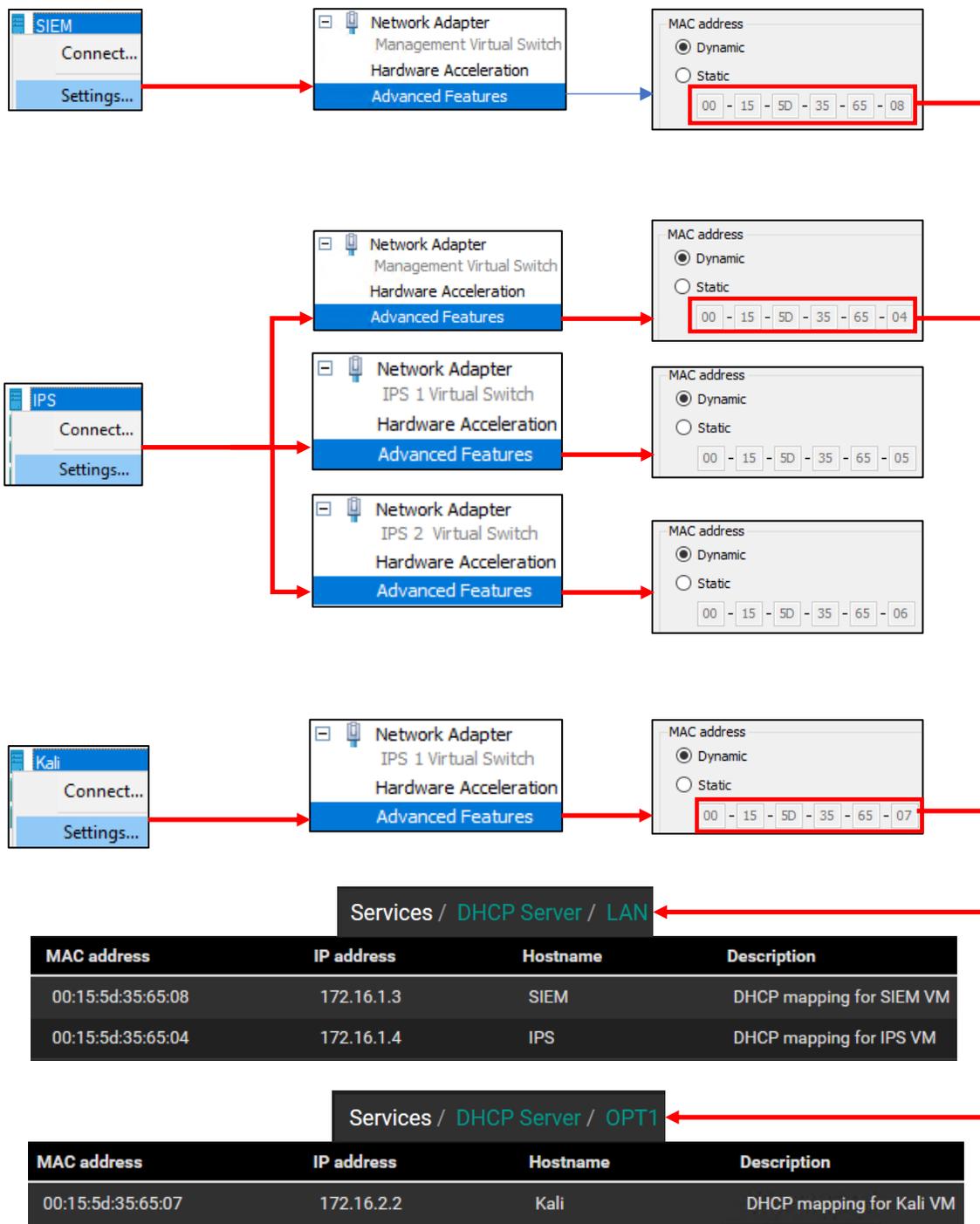
9-31: Navigate to the *Advanced Features* of the Kali VM's only network adapter and under the *Port mirroring* section, change its *Mirroring mode* setting to *Source*. For the IPS VM, students will need to create two more network adapters using the *Add Hardware* option. Attach one to the *IPS 1 Virtual Switch*, and the other to the *IPS 2 Virtual Switch*. Open the *Advanced Features* menu for these network adapters and under *MAC address*, select the *Enable MAC address spoofing* checkbox, and finally, set the mirroring mode for both new adapters to *Destination*. As always, Click *Apply* to confirm these configuration changes, then *OK* to exit.

Before we proceed, there's one more adjustment students need to make to their virtual machines: the SIEM, IPS, and Kali VMs all need to be powered on and powered off again. Recall in [section 9.4.4](#) (pp. 117-119) that Client Hyper-V doesn't assign MAC addresses to network adapters until after the virtual machine is first booted. Students will power on, then immediately turn off all three virtual machines. This will force Hyper-V to assign MAC addresses to each virtual adapter on each virtual machine.

Students will then record the MAC addresses of each virtual adapter, the virtual machine they belong to, and the virtual switch they are attached to. From there, students will log into the pfSense VM and configure static DHCP allocations for the SIEM VM, Kali VM, and the IPS VM (but *just* the adapter attached to the *Management Virtual Switch*). The SIEM VM should be statically assigned the IP address 172.16.1.3, the *Management Virtual Switch* interface of the IPS VM should be assigned 172.16.1.4. Both of these allocations should be configured on the *LAN* interface of the IPS VM. Meanwhile, the Kali VM should be assigned the IP address 172.16.2.2 on the *OPT1* interface. Students may refer to Chapter 14, [section 14.3.4.1](#) (pp. 690-692) for a refresher on creating static DHCP mappings on pfSense.



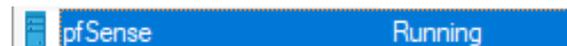
9-32: Turn the SIEM, IPS, and Kali VMs on and off again to force Hyper-V to assign MAC addresses to the network adapters. Students can select each virtual machine individually then right click the *Start*, then *Turn Off* options, or they can select multiple virtual machines at once by holding the ctrl key, and left-clicking each VM they would like the *Start* and *Turn Off* actions to apply to.



9-33: With the SIEM, IPS and Kali VMs powered off again, open up their individual settings menus and open up the *Advanced Features* sub-menu for each virtual machine's attached network adapter. Record the MAC address for each interface, and note which virtual switch its attached to (check out the *Noting the Notable* sidebar conversation on page 119 for a nice template to use for documenting the lab VMs). Use the MAC addresses and create three static DHCP mappings on the pfSense webConfigurator. Create two mappings on the *LAN* interface for the SIEM VM, and the *Management Virtual Switch* interface of the IPS VM, and a third mapping on the *OPT1* interface for the Kali VM. Be sure to *Apply Changes* on the pfSense webConfigurator to save these new DHCP mappings.

9.5.2 Operating System Installation

In this section, students will learn how to install the operating system for the SIEM, IPS, and Kali virtual machines. Both the SIEM and IPS VMs will have Ubuntu Server installed as their operating system, while the Kali VM will have the latest version of the Kali Linux distribution installed. As a general reminder, students should make sure that the pfSense VM is running, and that they have completed chapter 14 to ensure pfSense is ready to support the rest of the lab environment. To confirm the pfSense is running, students can check the *State* column on the *Virtual Machines* section of the *Hyper-V Manager* window. The *State* field for the pfSense VM should have the text *Running* if the virtual machine is active.



9-34: The *Hyper-V Manager* window can be used to confirm that the pfSense VM is running.

9.5.2.1 Installing Ubuntu on the SIEM VM

To get started, right-click the SIEM VM in the *Hyper-V Manager* window, and select *Start* to power on the SIEM virtual machine, then right-click again, and select *Connect* open up the SIEM VM's virtual console. The virtual machine will begin booting off the Ubuntu Server ISO. The first screen will ask students to confirm the language you wish to use. The default language should be *English*, so hit the enter key on your keyboard to continue.

Depending on when students downloaded their copy of the Ubuntu Server ISO, and how frequently the ISO is updated, a screen may appear titled *Installer update available*. This screen provides users with the option to download the latest version of the Ubuntu installation wizard, called Subiquity. Use the arrow keys on your keyboard to highlight *Update to the new installer*, then hit enter.

Note: If for some reason downloading the latest installer fails, there's a good chance that there are network problems with the lab environment elsewhere, and that there is troubleshooting to do. Students are welcome to select the *Continue without updating* option, but keep this in mind if the installer misbehaves or fails later. Check to see if the hypervisor host has internet connectivity, double check the firewall rules on the pfSense virtual machine, network settings, physical cabling, etc.

The next screen asks users to confirm their keyboard configuration. The default settings for both the *Layout* and *Variant* settings are *English (US)*. If you are not using a standard US-English keyboard, you may wish to use the arrow keys to highlight the *Identify keyboard* option, then hit enter. Otherwise, highlight *Done* on the bottom of the screen, and hit enter.

Next up, is the *Network connections* screen. A single network adapter should populate this page. The network adapter (named `eth0` in my case) should automatically be assigned the IP address 172.16.1.3. Below the IP address in light grey text is the MAC address of the network adapter that the Ubuntu installer detected. This should be the same MAC address of the network adapter of the SIEM VM. If the correct IP address was assigned, students can hit the enter key to continue (The *Done* option should be highlighted by default). Otherwise, see the sidebar conversation, *What Reservation?* for some troubleshooting tips.

What Reservation?

If for some reason the network adapter was assigned any other IP address other than 172.16.1.3, check the MAC address of network adapter of the SIEM VM. Compare that MAC address to the MAC address used to create a static DHCP mapping on the *LAN* interface of the pfSense VM in [section 9.5.1](#) (pp. 135-140). Compare that to the MAC address displayed on the *Network connections* screen of the Ubuntu installer. ***They should all be identical*** (missing colon [:] symbols notwithstanding). Correct any errors with the pfSense static DHCP mapping, and reset the SIEM VM to restart the Ubuntu installer until the pfSense DHCP assigns the network adapter the correct IP address.



The image shows three screenshots from a network configuration process. The first is a 'Static' configuration form with a red box around the MAC address fields containing '00 - 15 - 5D - 35 - 65 - 08'. The second is a table with 'MAC address' and 'IP address' columns, with a red box around the MAC address '00:15:5d:35:65:08' and the IP address '172.16.1.3'. The third is a terminal output showing network details for 'eth0' with a red box around the MAC address '00:15:5d:35:65:08'.

MAC address	IP address
00:15:5d:35:65:08	172.16.1.3

```
[ eth0 eth -  
DHCPv4 172.16.1.3/24  
00:15:5d:35:65:08 / Ur
```

9-35: If the SIEM VM failed to get the correct IP address, check the *Advanced Features* menu of the network adapter in the *Hyper-V Manager*, the MAC address used to create a static DHCP mapping on the *LAN* interface on the pfSense WebConfigurator. Correct the static DHCP entry as necessary then restart the SIEM VM to restart the ubuntu installer. Confirm that the network adapter was correctly assigned the 172.16.1.3 IP address.

The *Configure proxy* screen appears. Use the up arrow key to highlight the text box labeled *Proxy address* and enter `http://172.16.1.1:3128`. If you recall from Chapter 14, this is the IP address and port for the Squid proxy service on the *LAN* interface of the pfSense VM. Use the arrow keys to highlight *Done*, and hit enter to continue.

The next screen, labeled *Configure Ubuntu archive mirror* will appear. This is another one of those situations where students will know whether or not they need to change this setting. Unless the lab environment is in an enterprise network and the network team happens to be operating their own software archive mirror, accept the default setting (in my case, the default mirror address was `http://us.archive.ubuntu.com/ubuntu`). With *Done* highlighted, hit enter to continue.

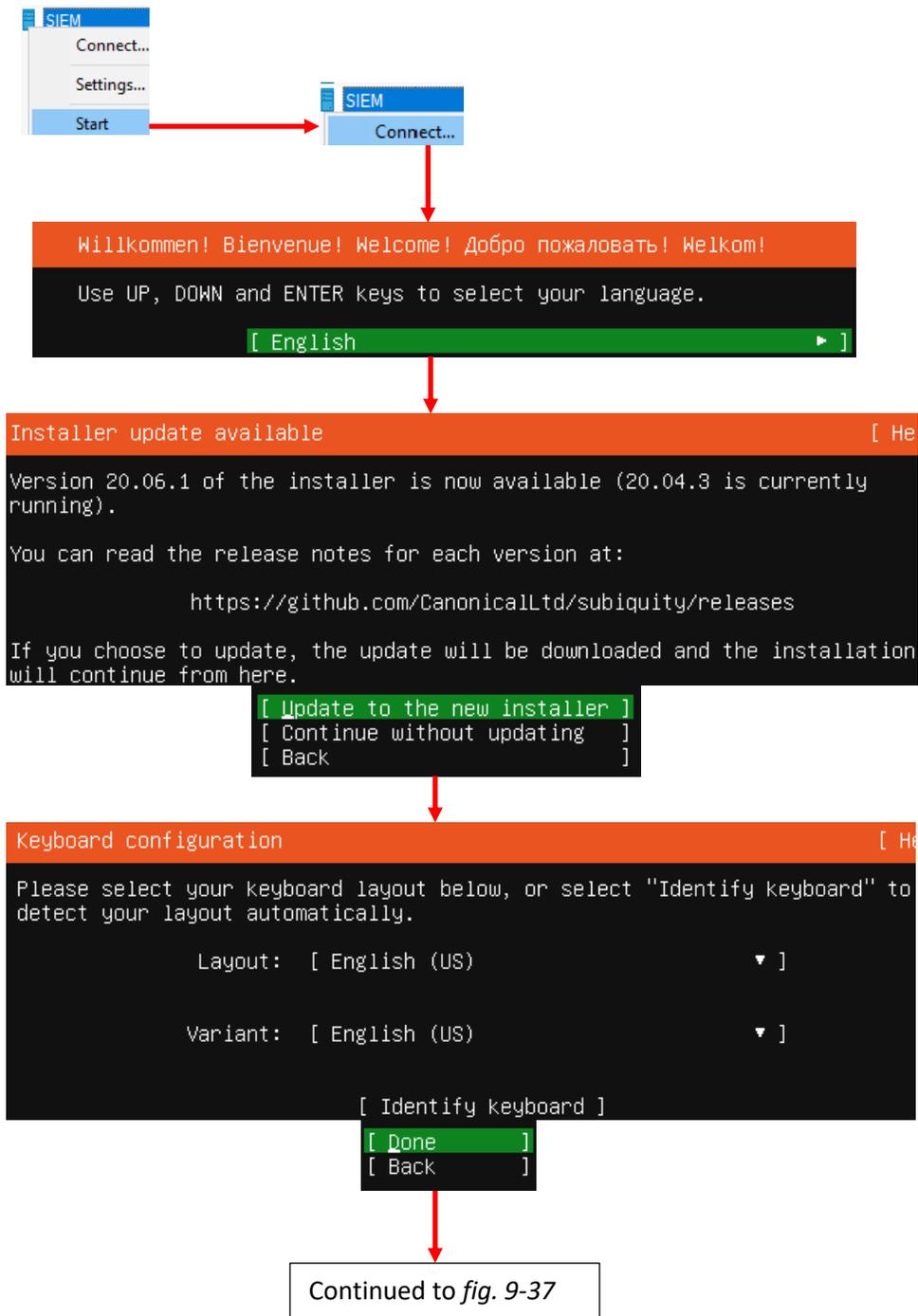
The *Guided storage configuration* screen appears next. Accept the default settings and let the Ubuntu installer format the entire disk. Use the arrow keys to highlight *Done*, and hit enter to continue. The next screen, titled *Storage configuration*, shows you how the installer is going to format the hard drive, and what partitions are going to be where in a large list labeled *FILE SYSTEM SUMMARY*. By default, *Done* should already be highlighted on this screen. If not, use the arrow keys to highlight it and hit enter to continue. A pop-up labeled *Confirm destructive action* appears. This screen informs the user that any data on the disk will be lost as a result of formatting and partitioning the disk. Since there is no data on the virtual hard disk yet, highlight *Continue*, then hit the enter key to proceed.

Next is the *Profile setup* screen. There are five input boxes on this screen. Ubuntu asks the user for their name, the server's name, a username (that will be used to log in to the server later), the password for that username, followed by an input box asking the user to repeat the password. Students may enter any name, username, or password they would like, but it is recommended to set the server's name to *siem*, and save the username and password combination for the SIEM VM to a password manager. Once finished, use the arrow keys to highlight *Done*, then hit enter to continue.

The *SSH Setup* screen appears and asks users if they would like to install the OpenSSH server package by default. By default, the prompt should be between two brackets next to the text *Install OpenSSH server*. Hit the spacebar to leave an 'X' between the brackets. Afterwards, use the arrow keys to highlight *Done* and hit enter to advance.

The next screen is labeled *Featured Server Snaps*. The latest versions of Ubuntu use an additional software manager called 'snap' to deliver software packages. Use either the arrow keys or the tab key to highlight *Done* and hit enter – Do not install any snaps, continue the installer.

We reach a new screen labeled *Install complete!* At this point, students have made all of the necessary decisions for the Ubuntu installer to proceed, and handle all of the installation tasks at once. Once completed, the installer will grant students the option to reboot the system. However, instead of using the installer's reboot function, select the *Turn Off* option on the virtual console's navigation menu to shut down the SIEM VM.



9-36: In the Hyper-V Manager, right click on the SIEM VM listing, select *Start*, then right click again and select *Connect* to open the virtual console. The Ubuntu installer, Subiquity, should appear and ask students for their preferred language. Next, the installer checks for updates for itself, then asks the user to set the keyboard layout and variant language.

Continued from *fig. 9-36*

```
Network connections [ Help ]
Configure at least one interface this server can use to talk to other machines, and which preferably provides
sufficient access for updates.

NAME    TYPE  NOTES
[ eth0  eth  -      ]
DHCPv4  172.16.1.3/24
00:15:5d:35:65:08 / Unknown Vendor / Unknown Model
```

```
[ Done ]
[ Back ]
```

```
Configure proxy [ Help ]
If this system requires a proxy to connect to the internet, enter its details
here.

Proxy address: http://172.16.1.1:3128
If you need to use a HTTP proxy to access the outside world,
enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of
"http://[[user]][:pass]@]host[:port]/".
```

```
[ Done ]
[ Back ]
```

```
Configure Ubuntu archive mirror [ Help ]
If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address: http://us.archive.ubuntu.com/ubuntu
You may provide an archive mirror that will be used instead of
the default.
```

```
[ Done ]
[ Back ]
```

Continued to *fig. 9-38*

9-37: The next stages of the Ubuntu Server 20.04 installer. In these screens, students can confirm whether or not the static DHCP mapping for the SIEM VM is working correctly, configure the system to use the Squid proxy service configured on the pfSense VM, and confirm software archive mirror they would like to use.

Continued from *fig. 9-37*

```
Guided storage configuration
Configure a guided storage layout, or create a custom one:
(ⓧ) Use an entire disk
    [ VBOX_HARDDISK_VBc100006d-4f1fd0c6 local disk 80.000G ▾ ]
[X] Set up this disk as an LVM group
    [ ] Encrypt the LVM group with LUKS
        Passphrase:
        Confirm passphrase:

[ Done ]
[ Back ]
```

```
Storage configuration [ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT      SIZE      TYPE      DEVICE TYPE
[ /              39.498G   new ext4  new LVM logical volume ▶ ]
[ /boot         1.000G   new ext4  new partition of local disk ▶ ]

AVAILABLE DEVICES
DEVICE                                     TYPE      SIZE
[ ubuntu-vg (new)                          LVM volume group  78.996G ▶ ]
free space                                 39.498G

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES
DEVICE                                     TYPE      SIZE
[ ubuntu-vg (new)                          LVM volume group  78.996G ▶ ]
ubuntu-lv    new, to be formatted as ext4, mounted at /  39.498G ▶ ]
[ VBOX_HARDDISK_VBc100006d-4f1fd0c6        local disk  80.000G ▶ ]
partition 1  new, bios_grub                                1.000M ▶ ]
partition 2  new, to be formatted as ext4, mounted at /boot  1.000G ▶ ]
partition 3  new, PV of LVM volume group ubuntu-vg          78.997G ▶ ]

[ Done ]
[ Reset ]
[ Back ]
```

Continued to *fig. 9-39*

9-38: These screens are used to configure the storage settings for the operating system. Students will be using the default storage settings for the SIEM VM.

Continued from *fig. 9-38*

```
Confirm destructive action

Selecting Continue below will begin the installation process and
result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the
installation has started.

Are you sure you want to continue?

[ No ]
[ Continue ]
```

```
Profile setup [ Help ]

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

Your name: ayy
Your server's name: siem
The name it uses when it talks to other computers.
Pick a username: ayy
Choose a password: *****
Confirm your password: *****

[ Done ]
```

Title:	SIEM VM
Username:	ayy
Password:	*****
URL:	172.16.1.3
<input type="checkbox"/> Expires:	7/21/2020 12:11 PM
<input checked="" type="checkbox"/> Notes:	User account credentials for the SIEM VM.

Continued to *fig. 9-40*

9-39: After confirming the storage configuration settings, users are prompted to name their server, and create a user account. It's recommended to store the username and password for the SIEM VM in a password manager.

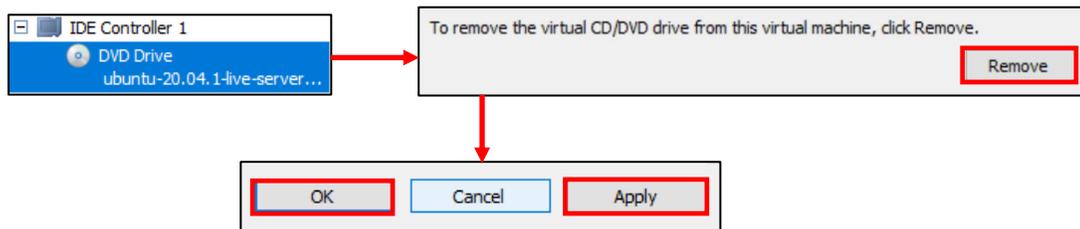
Continued from *fig. 9-39*



9-40: The final stages of the Ubuntu Server 20.04 installer. Select the option to install OpenSSH server, then decline to install any server snaps. Finally, once the installer grants you the option to reboot, ***Turn Off the virtual machine via the virtual console's navigation menu.***

9.5.2.2 Additional Virtual Machine Settings – SIEM VM

Now that the operating system installation is complete, there is one last configuration setting to adjust on the SIEM VM before we can boot into Ubuntu Linux and perform some diagnostic tasks. Back in [section 9.4.4](#) (pp. 117-119), students learned how to remove the *DVD Drive* virtual hardware from the pfSense VM. That task needs to be performed on the SIEM VM. Open up the SIEM VM's *Settings* menu, locate the *DVD Drive* item (On my VM, it was located under *IDE Controller 1*, but this may differ), left click on it to highlight it, then click the *Remove* button on the right pane. Afterwards click the *Apply*, then *OK* buttons in the lower right corner to confirm this change, and close the SIEM VM's settings menu.



9-41: Access the SIEM VM's settings menu and remove its *DVD Drive* virtual hardware.

9.5.2.3 Booting the SIEM VM for the first time

After changing the SIEM VM's settings, start the VM back up and bring up its virtual console. After a moment or two, students will be greeted with login prompt labeled *SIEM login*. Students should enter the user name and password configured while installing Ubuntu Server on the SIEM VM. After logging in, run the following commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The purpose of the `ip` command above is to display all of the network interfaces on the system. We pass this command the `-br` option for brief output, followed by the letter 'a' to indicate we're interested in seeing the IP (a)ddresses on our system. Users could replace 'a' with 'address' or 'addr' and the `ip` command would interpret it the same. We're using this command to serve as a secondary confirmation that the SIEM VM was successfully assigned the IP address 172.16.1.3, as displayed in [fig. 9-42](#) below. Students may notice a second interface on the system designated `lo`. This is the "loopback" network interface and can safely be ignored.

The `nslookup` command is to confirm that the SIEM VM is able to resolve hostnames using DNS. The output from the command should be similar to what is presented in [fig. 9-42](#). Finally, that brings us to the `curl` command. This command is to confirm connectivity to the internet over port 443, HTTPS. The `-I` option in the command tells `curl` to only return the headers from the web server being contacted. Once again, the output of this command should be fairly similar to what is presented in [fig. 9-42](#).

```
Ubuntu 20.04 LTS siem tty1
siem login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8 ::1/128
eth0              UP                172.16.1.3/24 fe80::215:5dff:fe35:6508/64
ayy@siem:~$ nslookup www.google.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:806::2004
ayy@siem:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 21 Jul 2020 20:10:38 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 21 Jul 2020 20:10:38 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-21-20; expires=Thu, 20-Aug-2020 20:10:38 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=1AAB5nk21PEgo8rGiFr-9PxEuTIYONxZtMMi-EmACTdRnP1PkB0xoosGu9FjWzbLyW0TGOHKUj6kLonn4Rr-yu-MIc8itYAlSD7X2VkJk0UK30rSk6Fd0ce6_Battd9PQ4YI7-CoRGf38In74mD78YmTAAtz1XJf8-WM; expires=Wed, 20-Jan-2021 20:10:38 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

9-42: After logging in to the SIEM virtual machine, students will need to run a series of network troubleshooting commands. These commands are to confirm that the SIEM VM has the correct IP address configured (`ip -br a`), can resolve hostnames via DNS (`nslookup www.google.com`), and has connectivity over HTTPS (`curl -I https://www.google.com`).

Before logging out of the SIEM virtual machine, there are three more commands to run, but before we can run them, we will need to become the root user. Enter the following command:

```
sudo su -
```

When prompted, enter the password for the user account students are logged in as. If successful, students will be logged in as the root user on the SIEM virtual machine. The root user, sometimes referred to as the super user, is a special account that has complete authority over the system. Additionally, root has access to special administrative commands that normal users are not allowed to use. As the root user, **run the following three commands in this exact order:**

```
apt-get update
apt-get -y dist-upgrade
init 6
```

Ubuntu uses a package manager called apt (in addition to the snap package manager mentioned earlier). The two apt-get commands, apt-get update then apt-get -y dist-upgrade tell Ubuntu to reach out to the software archive mirror and get an updated list of software packages, then if any packages installed on the system need to be updated, updated them immediately. This set of commands also confirms that the Squid proxy server on the pfSense VM is working properly, and proxying all of the HTTP requests from the SIEM VM. The final command, init 6, tells the system to reboot immediately. As an alternative, users can also run the command reboot instead.

```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (524 kB/s)
Reading package lists... Done
root@siem:~# apt-get -y dist-upgrade
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@siem:~# init 6_
```

9-43: the command `sudo su -` allows students to become the root user on the SIEM VM. We then use root's permissions to ensure all of the software packages on the system are up to date (`apt-get update`, followed by `apt-get -y dist-upgrade`), then immediately reboot the system (`init 6`, or optionally `reboot`). Be aware that the apt-get commands may take a little bit of time to finish, based on the number of updates available and internet connection speed.

Help! My apt-get commands are failing!

If you're experiencing problems with the apt-get commands failing to complete, there's a very good chance that the apt package manager is not properly configured to use the SQUID HTTP proxy we installed on the pfSense VM, or that the SQUID proxy service on the pfSense VM may be misconfigured. If students entered the wrong information during the operation system installation (e.g., on the Configure Proxy screen during Ubuntu Server installation), then the apt package manager will not work properly.

Here are some troubleshooting steps to think about:

On the SIEM VM, run the command:

```
cat /etc/apt/apt.conf.d/90curtin-aptproxy
```

This command will read the contents of the file `/etc/apt/apt.conf.d/90curtin-aptproxy` and display its contents on the screen. The file should read something like this:

```
Acquire::http::Proxy "http://172.16.1.1:3128";  
Acquire::https::Proxy "http://172.16.1.1:3128";
```

If this file does not exist, or has any content that is in any way different from the lines above, **run the following three commands exactly as displayed, and in this exact order:**

```
sudo su -  
echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-  
aptproxy  
echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >>  
/etc/apt/apt.conf.d/90curtin-aptproxy
```

This series of commands requires root access, so the first thing we do is use `sudo su -` to become the root user. The next two commands delete the current `90-curtin-aptproxy` file if it exists, then overwrites it with the two correct entries that should exist in the file. After running these commands, run `cat /etc/apt/apt.conf.d/90curtin-aptproxy` once more, and confirm that the output matches the correct output listed above. After confirming that the configuration file has been recreated correctly, try running the apt-get commands once more. If they continue to fail, then continue the troubleshooting process. Assuming that the network connectivity check commands were successful (e.g., `nslookup` and `curl`), think about the following:

- Is the SQUID proxy service installed on pfSense?
- Is there a firewall rule on the LAN interface to allow access to the proxy service? (allow traffic to IP address 172.16.1.1 port 3128 TCP from network 172.16.1.0/24)
- Is the option *Resolve DNS IPv4 First* checked on the SQUID proxy service?

These are all configurations covered in chapter 14, and should have already been specified. Double check that they have been configured correctly, then try updating the SIEM VM again.

```

ayy@siem:~$ 1 sudo su -
root@siem:~# 2 echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 3 echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >> /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 4 cat /etc/apt/apt.conf.d/90curtin-aptproxy
Acquire::http::Proxy "http://172.16.1.1:3128";
Acquire::https::Proxy "http://172.16.1.1:3128";
root@siem:~# 5 apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [332 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [8,780 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [5,404 B]
Fetched 826 kB in 1s (972 kB/s)
Reading package lists... Done
root@siem:~# 6 apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@siem:~#

```

9-44: This illustration demonstrates how to fix or modify the `/etc/apt/apt.conf.d/90curtin-aptproxy` file, in the event that student find that there is a problem with the file. First utilize `sudo su -` (1) to become the root user. Then use the two `echo` commands (2, 3) to write the correct configuration data so that `apt` knows how and where to access the squid proxy configured on the pfSense VM. Utilize the `cat` (4) command to confirm that the configuration file is properly configured. Finally, run `apt-get update` (5) and `apt-get -y dist-upgrade` (6) to check for the latest updates and download them.

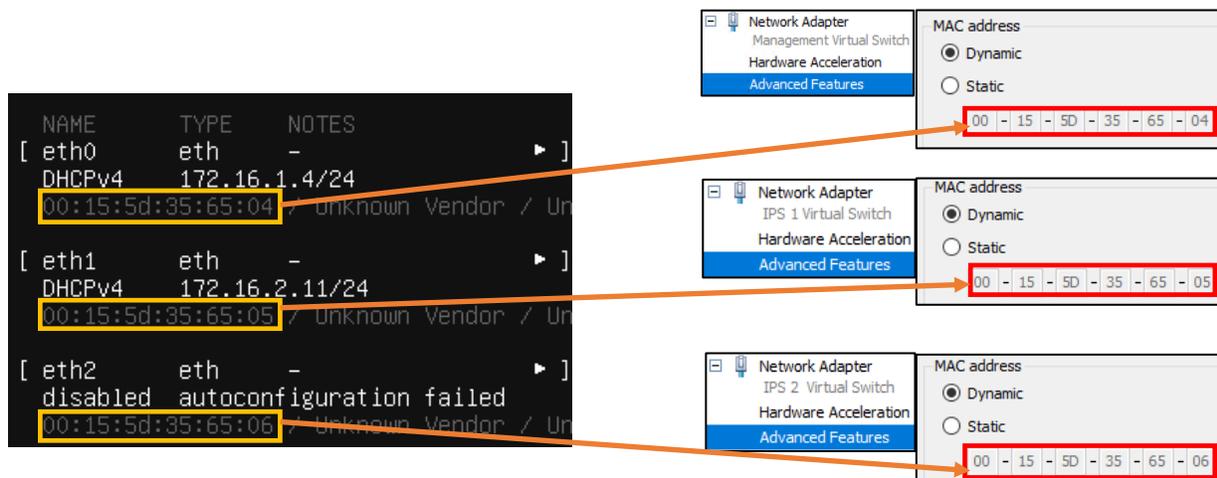
9.5.2.4 Installing Ubuntu on the IPS VM

Now that Ubuntu has been installed on the SIEM VM, network connectivity has been checked, and updates have been applied, next up is the IPS VM. The process for installing Ubuntu Server on the IPS virtual machine is practically identical, and will be summarized below:

- Start the IPS VM, and connect to its virtual console
- Select English as your language (or your preferred language)
- If there are any updates to Subiquity, select the option, *Update to the new installer*
- Select *English (US)* (or your preferred language) as the keyboard *Layout* and *Variant*

The *Network connections* screen will be a little bit different than it was on the SIEM VM, because the IPS virtual machine has three network interfaces. Recall in section 9.4.5, comparing and contrasting the MAC addresses of the three network adapters attached to the pfSense VM, and using that information to map the name of the network interface in pfSense (e.g., bridged virtual switch → em0/WAN, management virtual switch → em1/LAN, IPS 1 virtual switch → em2/OPT1).

Students will need to perform a similar exercise for the IPS virtual machine on the *Network connections* screen. In light grey text underneath the name of each network interface is the MAC address for that interface. Cross-reference the MAC address and interface name on the screen with the MAC address of adapters 1-3 recorded earlier. See *fig. 9-45* below for an example, based on the MAC addresses of my IPS virtual machine.



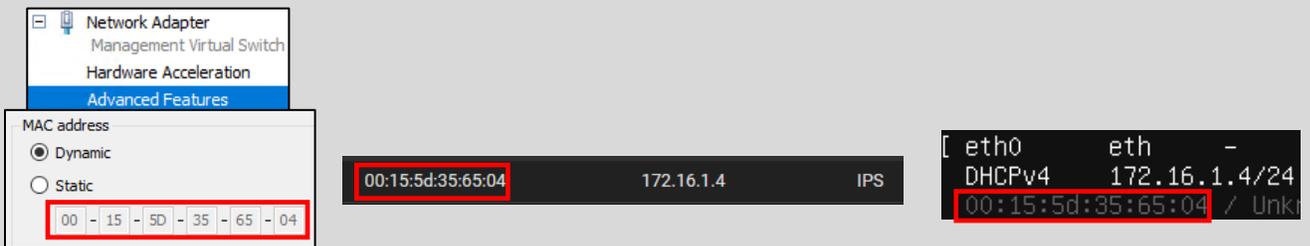
9-45: Ubuntu has assigned our three virtual adapters the interface names eth0, eth1, and eth2. Below each interface name is a MAC address. By cross referencing those MAC addresses to the MAC addresses and the virtual switches attached to the IPS VM, we can determine which interface name maps to which network adapter, and confirm which virtual switch the interfaces are attached to. For instance, eth0, is the network adapter attached to the management virtual switch. eth1 is attached to IPS 1 virtual switch, while eth2 is attached to the IPS 2 virtual switch.

Now that students are aware of which interface corresponds to which network segment, the next step is ensuring that the interface connected to the *Management Virtual Switch* (e.g., the LAN network in pfSense) is the only interface that has an IP address assigned. In [section 9.5.1, fig. 9-33](#), students created a static DHCP reservation for the IPS VM using the MAC address of the network adapter attached to the *Management Virtual Switch*, and assigned it the IP address 172.16.1.4. In [fig. 9-45](#), the interface eth0 has the IP address 172.16.1.4. This confirms students created the static DHCP allocation correctly on the pfSense WebConfigurator, and that eth0 is the interface connected to the *Management Virtual Switch*. If the network adapter attached to the LAN network does not have the correct IP address, there is a good chance that the static DHCP mapping for the IPS virtual machine is incorrect. Take a look at the sidebar discussion, *Reservation for One*, for some troubleshooting recommendations.

Reservation for One

If you're here, that means that the network interface attached to the LAN/Management network didn't get the IP address 172.16.1.4. Similar to the *What Reservation?* Sidebar discussion for the SIEM VM, you'll want to check a few things:

- Check the MAC address of the network adapter attached to the *Management Virtual Switch*
- Visit *Services > DHCP Server* on the pfSense webConfigurator, and check the static DHCP mappings of the *LAN* interface, particularly, the entry for the IPS VM
- Compare the MAC address of the previous two locations with the MAC addresses presented on the *Network Connections* screen. You should already know which interface name maps to which MAC address and virtual switch. In my case this was the interface eth0
- Make any necessary corrections to the static DHCP allocation for the IPS VM, then restart the VM. Make your way back to the *Network connections* screen, and confirm that the correct interface was assigned the correct IP address.

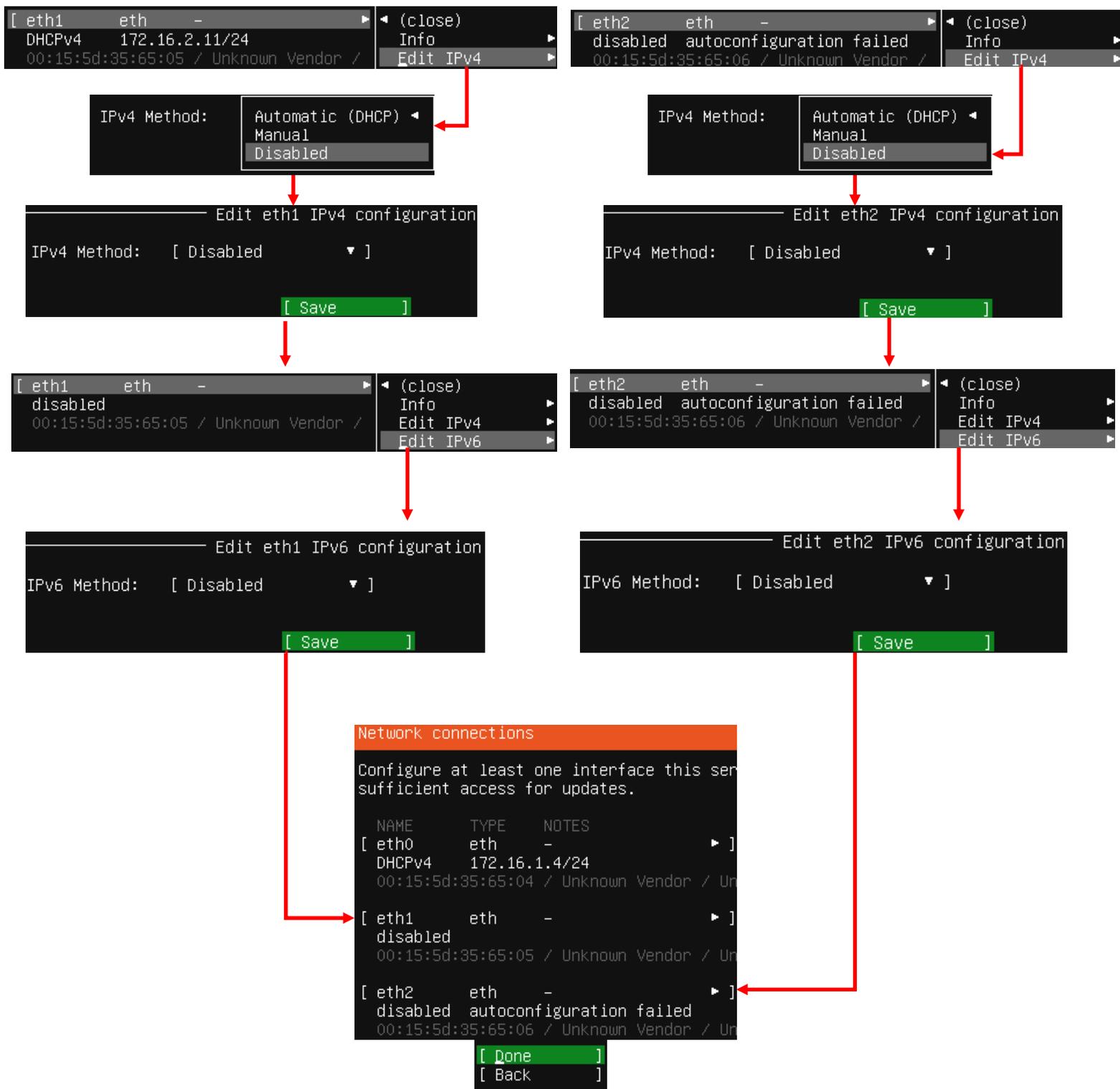


9-46: Just like with the SIEM VM, note the MAC address of the network adapter attached to the *Management Virtual Switch*. Compare that to the MAC address used to create a static DHCP mapping for the IPS VM on the LAN interface of the pfSense webConfigurator. If they don't match, correct the static DHCP mapping entry, then restart the IPS VM. Determine which interface was assigned the IP address 172.16.1.4.

The final step on the *Network connections* screen is to disable the remaining network interfaces. The interfaces connected to the IPS 1 and IPS 2 virtual switches (In the illustrations provided, these are the interfaces eth1 and eth2) should never receive an IP address. The lab environment, and IPS software we'll be using does not require these interfaces to have IP addresses, so we want to take advantage of that. Students may have notice that the interface connected to the *IPS 2 Virtual Switch* (eth2) doesn't have an IP address assigned, instead displaying the status: *disabled autoconfiguration failed*. Disregard this error message and follow the instructions below. Substitute the interface names eth1 and eth2 as necessary:

- Using the arrow keys, Highlight one of the other remaining interfaces. In my case, I chose to highlight eth1. Hit enter, and a dialogue box pops up.
 - Highlight the option *Edit IPv4*, and hit enter.
 - A new dialogue box appears titled *Edit eth1 IPv4 configuration*, with a single drop-down option highlighted, titled *IPv4 Method*. Hit enter again, and a list of choices appear. Use the arrow keys to select the option *Disabled*, and hit enter.
 - Use the arrow keys to highlight the option *Save*, and hit enter.
- Repeat the process again, only this time, Select *Edit IPv6*. By default, IPv6 should already be set to *Disabled*, but if it is not, follow the same process to set IPv6 to *Disabled*.
 - Highlight *Save*, and hit enter to exit the *Edit eth1 IPv6 configuration* dialogue box.
- Repeat this process for the final interface. In my case, eth2. Disable the IPv4 Configuration (in my case, it was already set to *Disabled*) and confirm that the IPv6 configuration is already *Disabled*.

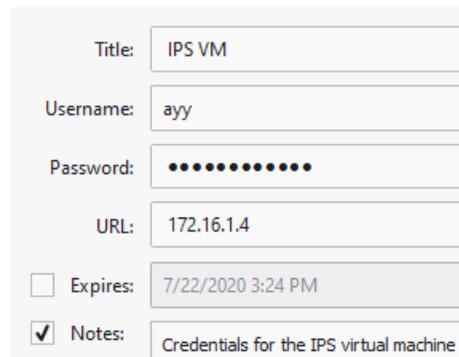
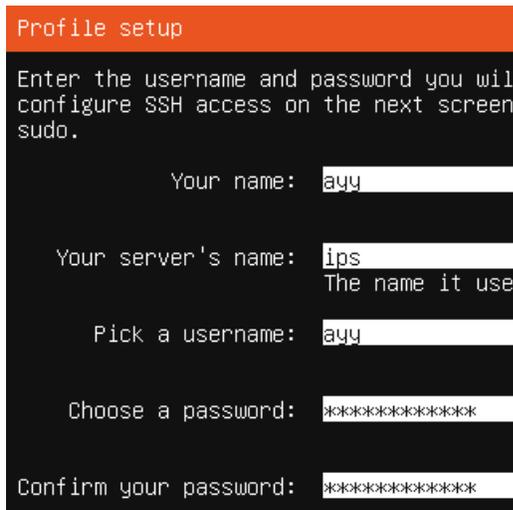
The end result should be one interface with the IP address 172.16.1.4, and two disabled network interfaces. Students can refer to *fig. 9-46* below for assistance. When finished, use the arrow keys to highlight *Done*, and hit enter to continue.



9-47: eth0 is the interface attached to the LAN network (management virtual switch), and should be the only interface with an IP address. ***Disable the other interfaces. They should never be assigned an IP address.***

The rest of the installation process for the IPS VM should be nearly identical to the SIEM VM:

- On the *Configure proxy* screen, set the *Proxy address* to `http://172.16.1.1:3128`
Accept the default archive mirror (or an alternative, if required) on the *Configure Ubuntu archive mirror* screen
- Accept the default settings on the *Guided storage configuration*, and *Storage configuration* screens. Select *Continue* on the *Confirm destructive action* dialogue pop-up
- Fill out the *Profile setup* screen, ensuring that the *Your server's name* input box is set to *ips*. Remember to document the username and password you create and store it in your preferred password manager

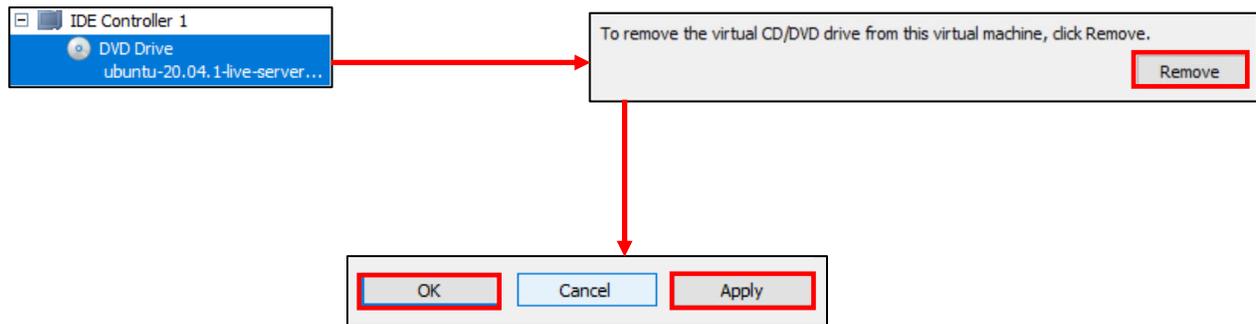


9-48: The *Profile setup* screen for the IPS virtual machine is, quite literally, the only other screen aside from the *Network connections* screen that differs from the installation process used on the SIEM VM. Be sure to save the username and password for the IPS VM to a password manager!

- On the *SSH Setup* screen, be sure to select *Install OpenSSH server*
- On the *Featured Server Snaps* screen, select *Done* and hit enter to move on to the *Installation complete* phase
- Once the installation has finished, use the *Turn Off* option on the virtual console's navigation menu to turn power off the IPS VM.

9.5.2.5 Additional Virtual Machine Settings – IPS VM

Now that Ubuntu Server is installed on the IPS VM, the only thing left to do is remove the DVD Drive virtual hardware. The process is identical used to remove the DVD drive from the SIEM Virtual machine – open the IPS VM's *Settings* menu, select the *DVD Drive* virtual hardware on the left pane, click the *Remove* button on the right pane, then click the *Apply* followed by *OK* buttons to save the virtual machine's configuration and exit the settings menu for the IPS VM.



9-49: Access the IPS VM's settings menu and remove its *DVD Drive* virtual hardware.

9.5.2.6 Booting the IPS VM for the first time

Start the IPS VM, Connect to its virtual console, and once Ubuntu has finished starting up and performing its first-time boot routines, log in with the username and password assigned on the *Profile setup* screen during the Ubuntu Server install. Once logged in, run the following three commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output from the `ip -br a` command will differ slightly, because the IPS VM has more network interfaces than the SIEM VM, but aside from that, the output from `nslookup` and `curl` should be more or less identical to the output of these commands from the SIEM VM. See *fig. 9-50* below for an example on what the output of these commands should look like.

```

Ubuntu 20.04 LTS ips tty1

ips login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ ip -br a
lo                UNKNOWN        127.0.0.1/8 :::1/128
eth0              UP            172.16.1.4/24 fe80::215:5dff:fe35:6504/64
eth1              DOWN
eth2              DOWN
ayy@ips:~$ nslookup www.google.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.8.196
Name:   www.google.com
Address: 2607:f8b0:4009:815::2004
ayy@ips:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Thu, 23 Jul 2020 17:21:51 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Thu, 23 Jul 2020 17:21:51 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-23-17; expires=Sat, 22-Aug-2020 17:21:51 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=No2uEqnF70q9zD3pzs0rY1b9m4o1HDDzP4BzZ1ULQM2ia7uXqWv97cWdZN0fc2JxijI_BXyxhRfuF2EEvFV50ssKkaJRIZPxm4TbIdfzAihP6aW6FsTqHu6Kif6j75q06iuFFU-UP0oA73r0ytPyD314nvxBKnu1_rqEmla0Sic; expires=Fri, 22-Jan-2021 17:21:51 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"

```

9-50: Just like with the SIEM VM, students will log in to the IPS virtual machine and run a couple of network diagnostic commands. The output from `curl` and `nslookup` commands should be more or less identical to the output on the SIEM VM, but the `ip -br a` command will produce a few more lines of content. Ignoring the `lo` (loopback) interface, there should be three interfaces. Only one of them should have the status of UP. That interface should be the interface assigned to the LAN/Management network, with the IP address 172.16.1.4.

After running those commands to confirm the IPS VM has been assigned the proper IP address, can resolve hostnames, and has internet connectivity, run the following commands in order to become the `root` user, install updates on the IPS VM (and confirm the Squid proxy server is proxying the IPS VM's HTTP requests), then reboot after the system is done installing those updates:

```

sudo su -
apt-get update
apt-get -y dist-upgrade
init 6

```

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [306 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [114 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [7612 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [136 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [4792 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Fetched 889 kB in 1s (1269 kB/s)
Reading package lists... Done
root@ips:~# apt-get -y dist-upgrade
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@ips:~# init 6

```

9-51: These commands are identical to the ones students ran on the SIEM virtual machine, and the serve the same purpose for the IPS VM: become the root user, check for updated packages, install those updates, then reboot the system.

Note: If you're having problems with your apt-get commands failing, refer back to the sidebar conversation on pp. 152-153, [*Help! My apt-get commands are failing!*](#) For further guidance. Students can follow the exact same steps laid out for the SIEM VM to troubleshoot the problem.

9.5.2.7 Installing Kali Linux on the kali VM

Now that the SIEM and IPS virtual machines are out of the way, next up is the kali VM. *Start* up the VM, then *Connect* to its virtual console. A boot menu appears with a number of options. Using the arrow keys, highlight *Install* and hit enter.

Similar to the Ubuntu installer, the first screen, titled *Select a language*, asks users to choose the language they want to use for their installation. The default setting is *English*, use the arrow keys to highlight another language as necessary, then hit enter. The next screen, *Select your location*, asks users to choose what country, territory or area in which they are located. This screen defaults to *United States*. Use the arrow keys to change this value as necessary, and hit enter to continue. Next up is the *Configure the keyboard* screen, that asks the user what keymap to use for their installation. The default setting is *American English* and can be changed with the arrow keys. After highlighting a keymap, hit enter to continue.

The installer begins loading other phases and components in order to continue the installation process. Afterwards, it will attempt to get an IP address. The pfSense DHCP server should give it an IP address through the OPT1 DHCP server, but students will not be able to confirm if the IP

address 172.16.2.2 was correctly assigned until after the operating system is installed. The next screen, titled *Configure the network*, prompts users to enter a hostname for the system. Students should use the default hostname *kali*. Hit the enter key to continue to the next screen that prompts for a domain name. Again, students may hit enter and accept the default, *localdomain*.

The *Set up users and passwords* screen appears. The first window asks for the full name of the user to be created. Type in the full name of the user account, and hit enter to continue to the next screen, that prompts for a username students will use to log in to the system. After typing in a username, hit enter to be prompted to create a password for this account. After hitting enter again, you'll be prompted to enter the same password again to confirm your choice. Enter the same password and hit enter to continue the installer. Just like with the SIEM and IPS virtual machines, students should save the username and password for the kali VM to their preferred password manager.

Next up is the *Configure the clock* dialogue. The installer will reach out to its preferred NTP servers to get the current time, then ask the user to select the time zone in which they are located. Use the arrow keys to choose a time zone, and hit enter to continue.

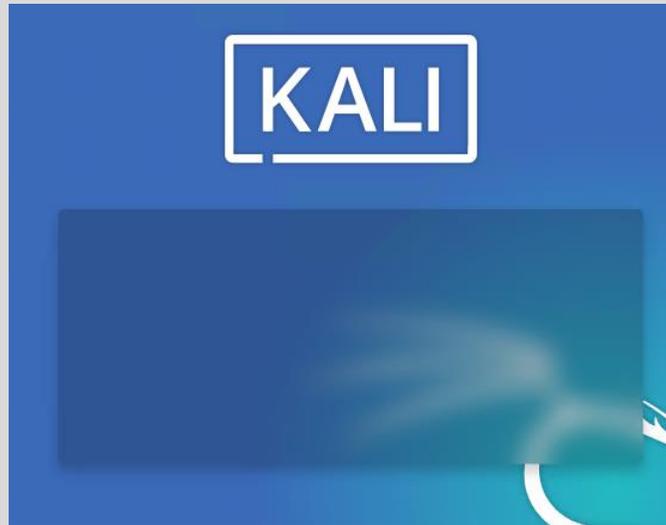
The *Partition disks* screen appears and asks users to select a partitioning method. Highlight the selection *Guided – use entire disk*, and hit enter. Users are then prompted to select the disk to partition. Since there is only a single virtual disk for the kali VM, hit enter to proceed. The next screen prompts students to select the partitioning scheme. Highlight the option *All files in one partition (recommended for new users)* and hit enter. Users are asked to confirm their choices on the next screen. Highlight the option *Finish partitioning and write changes to disk*, and hit enter. One final pop-up appears to annoy you, asking if students are sure they want to proceed, highlight *<Yes>* to confirm your choices, then press enter to continue.

The installer proceeds and begins installing the base operating systems components to the newly partitioned disk. After a moment or two, a window labeled *Software selection* appears. As the name implies, this screen allows users to pick additional software packages to install. Accept the default selections by pressing the tab key to highlight *<Continue>*, and hitting enter. The next portion of the installer retrieves and installs the requested packages. This portion of the installation may take some time, depending on internet speed and virtual machine performance.

After some time has passed a new prompt appears, labeled *Install the GRUB boot loader on a hard disk*, asking if users want to install the GRUB boot loader. This is a necessary component in order to boot the virtual machine, so highlight *<Yes>*, and press the enter key to continue. The next screen asks what partition to install the boot loader to. Seeing as how there is only one partition available, highlight it, and hit enter to proceed. After a moment or two passes, students are prompted to remove the installation media, and reboot the virtual machine to complete the installation. Just like with the SIEM and IPS virtual machines, *Turn Off* the virtual machine, then close the virtual console.

The Only Installer Menu that makes you Try Harder™

While I was in the midst of writing the virtual lab documentation for Client Hyper-V users, I experienced an extremely unusual bug: **Occasionally the kali installer menu fails to load.** The installer menu is the simple text-based menu that provides you with a variety of different installation options. Normally this installer is the first thing students will see when booting up and connecting to the Kali VM's virtual console. If you're experiencing this problem, the little background splash logo will appear, and the translucent textbox that normally contains the installer menu will appear, but there will be no installer menu options:



9-52: Has this happened to you? Here are a couple of work-arounds that may potentially fix it. In the meantime, I submitted a bug report to the distro maintainers to hopefully fix this issue permanently. Unfortunately, I have no idea what the root cause of this problem is.

So, how does one fix this issue? Over the course of trying a number of different things, here are some methods I learned that, for reasons that are entirely beyond my comprehension, have fixed this issue for me:

Method 1: *Pause and Resume* the Virtual Machine

On the Virtual Machine Connection's (e.g., the virtual console) navigation menu, there is a button that can be used to pause the virtual machine. This suspends the virtual machine's operations until it is unpaused by pressing the same button. Occasionally, I've found that if you pause then resume the virtual machine in rapid succession, this will sometimes force the installer menu to load. Sometimes, you only have to do this once to cause the installer menu to load, other times, I've done this 4-8 times in rapid succession before the installer menu decides it wants to appear and cooperate.



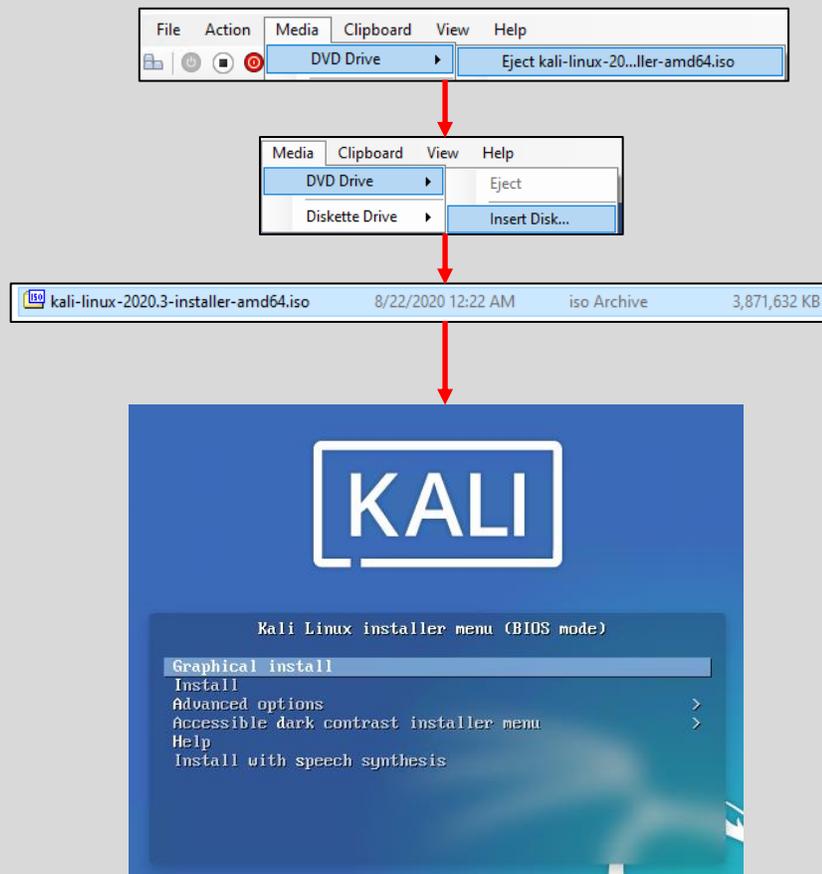
9-53: If the installer menu isn't loading for you, sometimes hitting *Pause* button on the Virtual Machine Console's navigation menu, followed by the *Resume* button in rapid succession will cause the installer menu to stop hiding from you. Sometimes, you only need to *Pause/Resume* once for this method to work. Other times, I've had to do this 6 (or more) times in a row before the installer menu would appear.

Method 2: Unmount and Re-mount the installation ISO

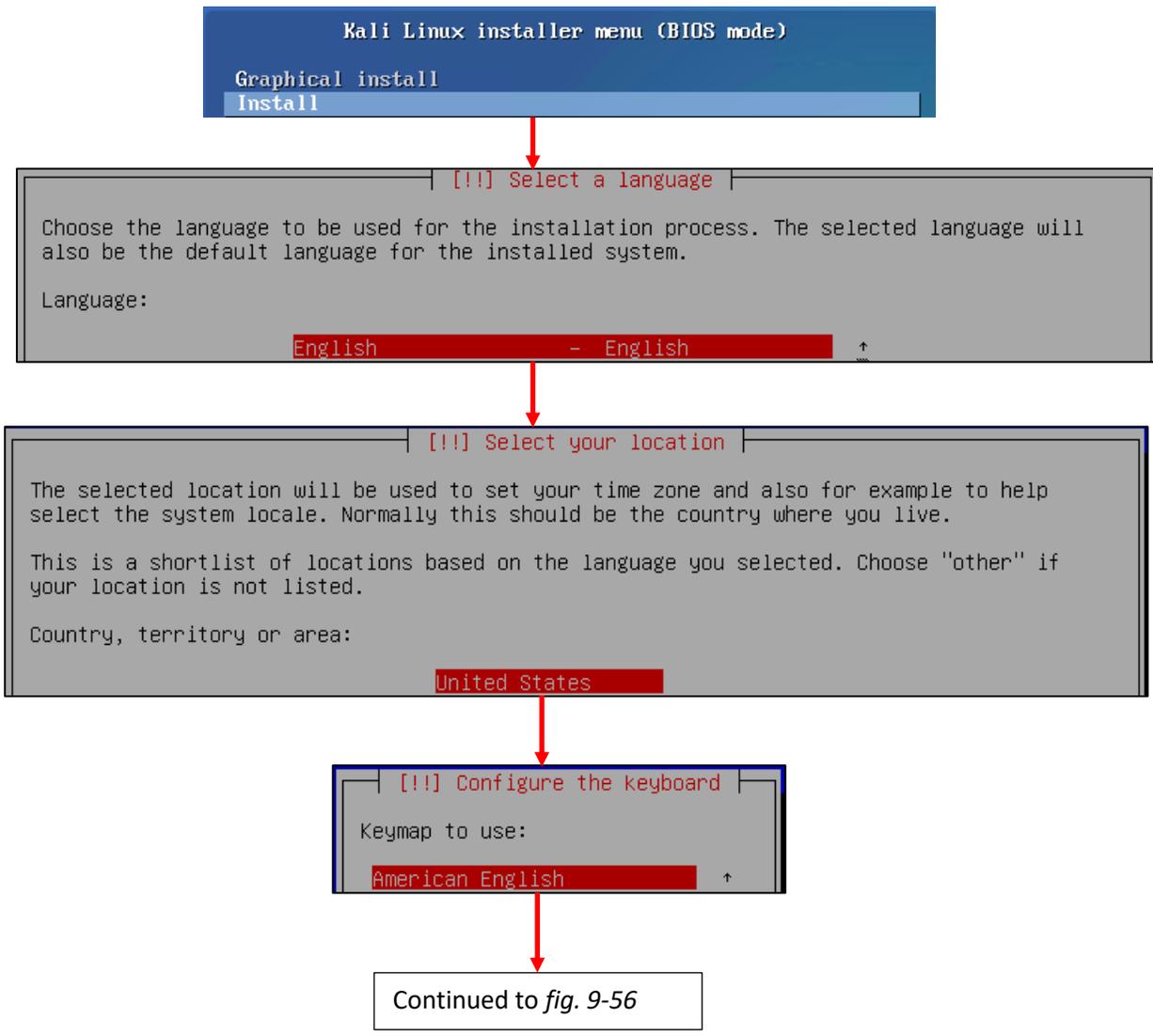
Sometimes after pausing/resuming the virtual machine, if you unmount and re-mount the Kali Linux installation ISO, this will occasionally cause the installer menu to appear as well. Follow the instructions from method 1 above – Click *Pause*, then *Resume* on the virtual machine console's navigation menu. Afterwards, click on the menu option labeled *Media*, then click on the *DVD Drive* option that appears in the drop-down list. Click on the option labeled *Eject*, followed by the name of the Kali Linux installation ISO file to unmount it. This is the equivalent of pressing the eject button on a CD/DVD drive. Next, click on the *Media* menu again, select *DVD Drive*, then select the *Insert Disk* option. Navigate to your Kali Linux ISO again and select it to re-mount it. This is like closing the CD/DVD drive.

Occasionally, I've seen this method cause the installer menu to load as soon as the ISO file is unmounted/ejected. Other times, I've seen the menu load after the ISO was re-mounted/inserted, and still in other cases, I've seen this method *not work at all*. **If this method did not work for you, fall back to using method 1 – pausing/resuming the virtual machine repeatedly to attempt to get the installer menu to load.** It may take a couple of times, but method 1 always works eventually (at least thus far).

Again, I don't know the root cause of this problem, and I have only seen this problem occur with Client Hyper-V. However, I've documented it, and submitted a bug to the Kali Linux distro maintainers (<https://bugs.kali.org/view.php?id=6679>). Hopefully, we'll get a root cause and resolution out of this. In my experience, after the installer menu loads, the rest of the operating system installation process works perfectly fine



9-54: First, Pause/Resume the virtual machine, as described in method 1 (fig. 9-52). Then, select *Media > DVD Drive > Eject kali-linux...-amd64.iso* to unmount the installer ISO. Next, select *Media > DVD Drive > Insert Disk*. Browse to the Kali Linux installer ISO and select it to re-mount it. This is the equivalent of opening, then closing a physical computer's CD/DVD drive. With any luck, this may cause the installer menu to appear again. If it doesn't, fall back to method 1 and repeatedly pause/resume the virtual machine until the installer menu options appear.



9-55: The first screens have users select their preferred language, location, and keyboard keymap.

Continued from *fig. 9-55*

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

kali

<Go Back> <Continue>

Domain name:

localdomain

<Go Back> <Continue>

Continued to *fig. 9-57*

9-56: The next few screens configure some of the network settings. Students are prompted to enter a hostname and domain name. Students should use the default hostname of *kali*, and default domain name of *localdomain*.

Continued from *fig. 9-56*

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

ayy lmao

<Go Back> <Continue>

Username for your account:

ayy

<Go Back> <Continue>

Choose a password for the new user:

[] Show Password in Clear

<Go Back> <Continue>

Re-enter password to verify:

[] Show Password in Clear

<Go Back> <Continue>

Continued to *fig. 9-58*

Title:	kali VM
Username:	ayy
Password:	*****
URL:	172.16.2.2
<input type="checkbox"/> Expires:	7/24/2020 11:35 AM
<input checked="" type="checkbox"/> Notes:	credentials for the kali VM

9-57: Similar to the *Profile Setup* screen in the Ubuntu installer, the Kali Linux installer features a series of prompts to create a user account for the system. Students should save the credentials to their preferred password manager when finished.

Continued from *fig. 9-57*

[!] Configure the clock

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

Eastern

[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk

Select disk to partition:

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

Partitioning scheme:

All files in one partition (recommended for new users)

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

#1	primary	81.6 GB	f	ext4	/
#5	logical	4.3 GB	f	swap	swap

Undo changes to partitions
Finish partitioning and write changes to disk

Write the changes to disks?

<Yes> <No>

Continued to *fig. 9-59*

9-58: After setting the time zone, students will have to configure the partitioning scheme for the install. The highlighted options above should be selected by default. If not, use the arrows to select them, and press enter to continue.

Continued from *fig. 9-58*

```
[!] Software selection

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

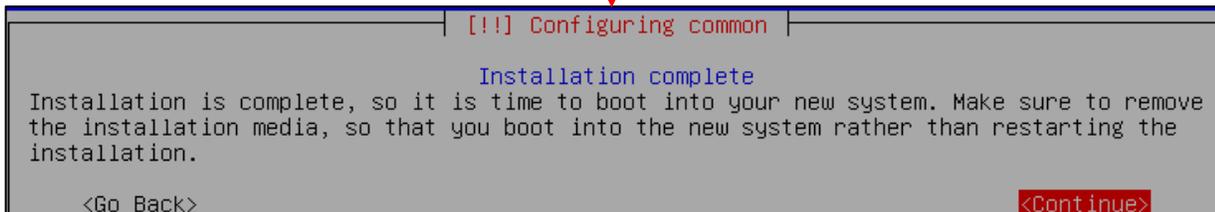
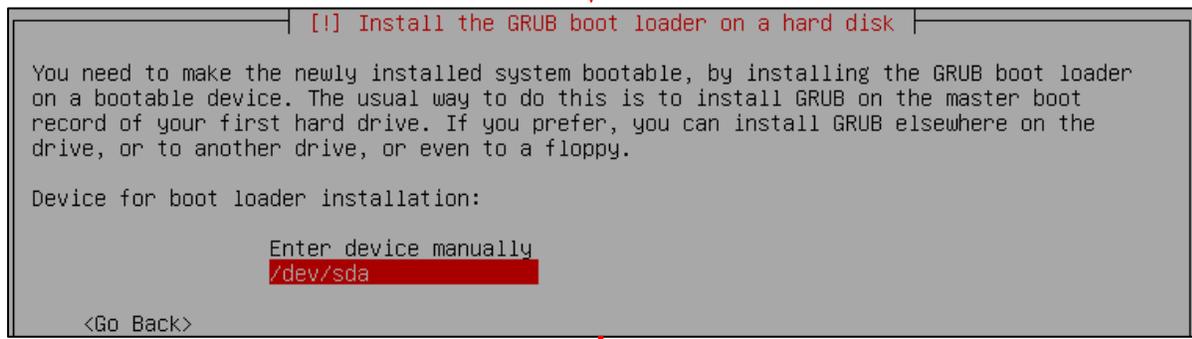
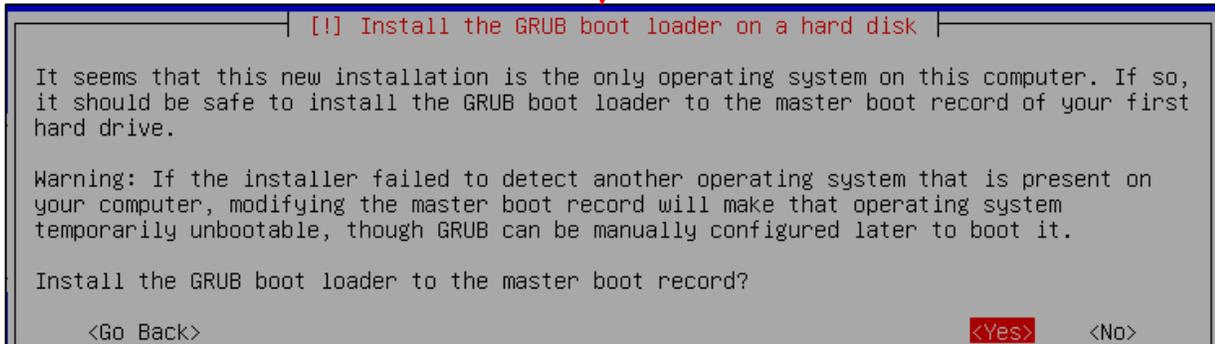
[*] Desktop environment [selecting this item has no effect]
[*] ... Xfce (Kali's default desktop environment)
[ ] ... GNOME
[ ] ... KDE Plasma
[*] Collection of tools [selecting this item has no effect]
[*] ... top10 -- the 10 most popular tools
[*] ... default -- recommended tools (available in the live system)
[ ] ... large -- default selection plus additional tools

<Continue>
```

Continued to *fig. 9-60*

9-59: On the *Software selection* screen, press the tab key to highlight *<Continue>*, and accept the default packages.

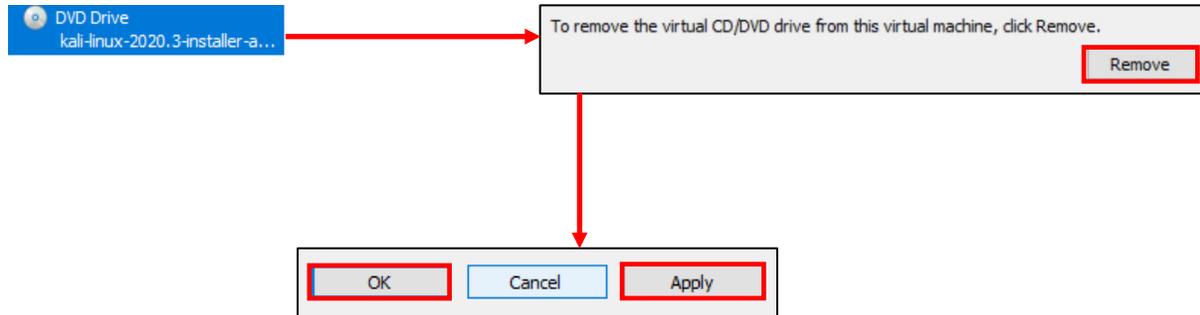
Continued from *fig. 9-59*



9-60: The final steps of the installation process. Install the GRUB boot loader to the only available disk on the system (this should be `/dev/sda`), wait for the screen labeled *Configuring common* to appear, then *Turn Off* the virtual machine and close the virtual console.

9.5.2.8 Additional Virtual Machine Settings – kali VM

By this point, it should already be established routine that once students are finished installing the operating system on their virtual machine, the next step is to remove its virtual DVD drive. Open up the Kali VM's *Settings* menu, find the *DVD Drive* hardware, and *Remove* it, *Apply* those changes, then click *OK* to exit the settings menu.



9-61: In the Kali VM's settings menu, remove its DVD drive, apply those changes, then exit the settings menu.

9.5.2.9 Booting the kali VM for the first time

With those last-minute virtual machine settings applied, *Start* the Kali virtual machine, then *Connect* to its virtual console. After a moment or two passes, students will be greeted with a graphical interface, asking for a username and password to log in. Enter the username and password supplied during the operating system install, and click *Log In* to continue.

On the top of the graphical user interface, there should be a menu bar with a few icons displayed. One of those icons is a small black window. Click on that icon to open a terminal session on the kali VM. With the terminal window open, run the same three commands that we ran on the SIEM and IPS virtual machines in order to confirm network connectivity is working as intended:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output of `ip -br a` should confirm that only a single interface (again, ignoring the `lo` interface) is installed on the system. That interface should have the IP address 172.16.2.2. As with the SIEM and IPS virtual machines, if this is not the case, students should compare the MAC address of network adapter on the kali VM to the MAC address of the static DHCP mapping made on pfSense. **Make sure the mac addresses match, and that the mapping was created on the OPT1 interface.**

As with the SIEM and IPS VMs, `nslookup` confirms the ability of the kali VM to resolve hostnames through DNS, and the `curl` command verifies that the VM can make outbound internet connections over HTTPS. The output of these commands should be similar to the output displayed in *fig. 9-62* below.

While Kali Linux is slightly different from Ubuntu, we can still use *most* of the same commands utilized on the SIEM and IPS virtual machines to become `root`, check for updates, then reboot the system. However, because we have no ability to define a proxy during the operating system installation phase, we also have to handle that task as well. **Run these commands in this exact order:**

```
sudo su -
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
apt-get update
apt-get -y dist-upgrade
init 6
```

Students may have noticed two new commands have been added here:

```
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
```

These commands are responsible configuring the apt package manager to use our HTTP proxy at 172.16.2.1:3128 on the *OPT1* interface of the pfSense VM. This is done by running the `echo` command, and redirecting its output (the `>` symbol) to the file `/etc/apt/apt.conf.d/99local` (a configuration file that the package manager will read when we run `apt-get` later). The second command, `cat /etc/apt/apt.conf.d/99local`, reads the contents of the file. If the output from the `cat` command reads:

```
Acquire::http::Proxy "http://172.16.2.1:3128";
```

Then that means apt was successfully configured to use the HTTP proxy. If the output from the `cat` command displays anything else, then students should re-enter the `echo` command.

Note: If most of these commands look familiar, it's because they're very similar to the troubleshooting commands I recommended in the sidebar discussion [*Help! My apt-get commands are failing!*](#) (pp. 152-153) for the SIEM and IPS virtual machines. There are a few key differences with the kali VM to be aware of, but for the most part, the troubleshooting steps laid out are the same as the steps I laid out in this section. Here are the key differences to be aware of:

- Make absolutely sure you are redirecting the output of the echo command to the file `/etc/apt/apt.conf.d/99local`. ***It must be that exact file, in that exact location.***
- The kali VM doesn't need the second line, `Acquire::https::Proxy "http://172.16.2.1:3128";`
- Make absolutely sure to specify the address of the OPT1 interface as the proxy address for the kali VM (e.g., `http://172.16.2.1:3128`).

After running these commands to configure the package manager, students should be able to run the remaining commands just like on the SIEM and IPS virtual machines. Bear in mind that Kali Linux is subject to frequent updates, and that some of those updates can be quite large. This means that depending on the performance of the Kali VM, and internet connection speeds, downloading and installing updates may take some time to complete.

```

ayy
.....
Cancel Log In

Terminal Emulator
Use the command line

ayy@kali:~$ ip -br a
lo UNKNOWN 127.0.0.1/8 ::1/128
eth0 UP 172.16.2.2/24 fe80::a9d9:817a:46ee:84c3/64
ayy@kali:~$ nslookup www.google.com
Server: 172.16.2.1
Address: 172.16.2.1#53

Non-authoritative answer:
Name: www.google.com
Address: 172.217.0.4
Name: www.google.com
Address: 2607:f8b0:4009:804::2004
ayy@kali:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 24 Jul 2020 19:55:47 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 24 Jul 2020 19:55:47 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-24-19; expires=Sun, 23-Aug-2020 19:55:47 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=rC1q-094PKdmAIZC2ajgCkpdGrGdulzdaxnJR2CUi-HYKBgbjzh_qCz6G5tJYoetE_Uc7rscR53x4Gri7HE3k_gu9h2BKh6etyF0hGDOiat3FF22oe-4VngjLFAdGEY3XTtecVHB8iJ5Qw2qUVjmmN7oZGeRUSj1mXJ8ulaLkRY; expires=Sat, 23-Jan-2021 19:55:47 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@kali:~$ sudo su -

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for ayy:
root@Kali:~# echo 'Acquire::http::proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
root@Kali:~# cat /etc/apt/apt.conf.d/99local
Acquire::http::proxy "http://172.16.2.1:3128";
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@kali:~# init 6

```

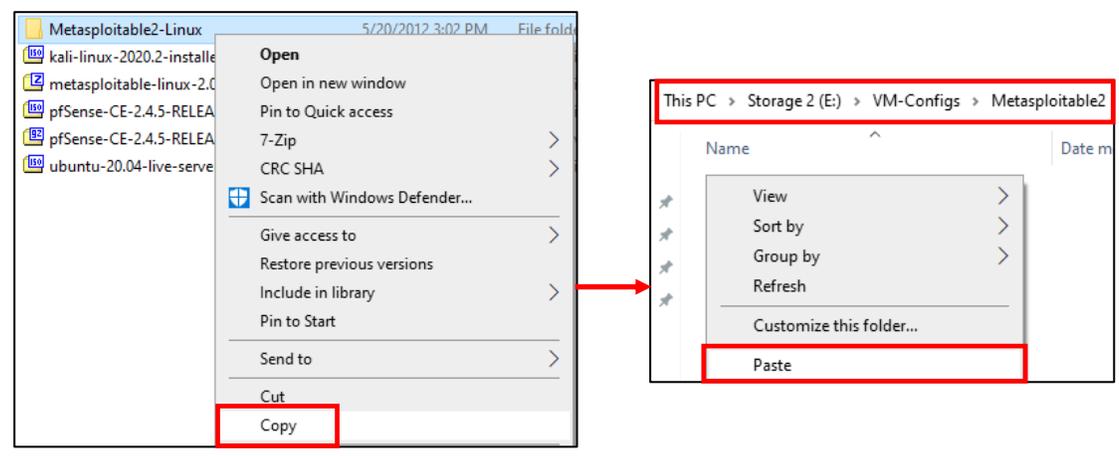
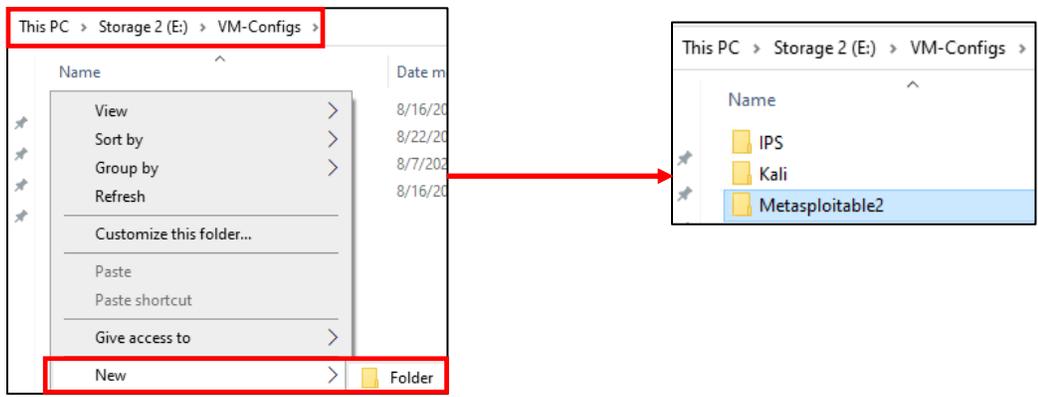
9-62: Login to the kali VM, configure the apt package manager to use the SQUID HTTP proxy on *OPT1* of the pfSense VM. Afterwards, install the latest operating system updates, then reboot the virtual machine.

9.5.3 Metasploitable 2

The Metasploitable 2 VM is slightly different than the other VMs, and will require a little bit of extra work, and some custom tools in order to get it up and running on Hyper-V. Before we can proceed, students must perform the following tasks:

-In [section 9.3.1](#) (pp. 92-95), students were instructed on how to access the *Hyper-V Settings* menu. Review the *Virtual Machines* setting. Create a subdirectory (New Folder) under this directory named Metasploitable2. Students will need to copy the decompressed Metasploitable 2 files to this directory. For example, if students are using the default Virtual Machines directory (C:\ProgramData\Microsoft\Windows\Hyper-V\), the full path with the newly created folder/subdirectory will be: C:\ProgramData\Microsoft\Windows\Hyper-V\Metasploitable2.

-Way back in chapter 1, [section 1.8](#) (pp. 33-35), students were instructed on how to use 7-zip to decompress files on Windows. Students should have already decompressed the Metasploitable 2 VM, metasploitable-linux-2.0.0.zip. This file, when decompressed, should have yielded the directory Metasploitable2-Linux. This directory needs to be moved or copied to the Metasploitable2 folder students created above. Right-click the Metasploitable2-Linux directory, and select *copy*. Then, navigate to the destination folder, right-click again, and select *paste*. See [fig. 9-63](#) for a demonstration.



9-63: Create a subdirectory named `Metasploitable2` under the *Virtual Machines* directory specified in [section 9.3.1](#) (pp. 92-95). In the illustration above, the *Virtual Machines* directory has been set to `E:\VM-Configs`, so the full path is `E:\VM-Configs\Metasploitable2`. Copy the decompressed `Metasploitable2-Linux` directory, and Paste it to this newly created subdirectory.

9.5.3.1 Converting the Metasploitable.vmdk to VHDX

Before students can create the Metasploitable 2 virtual machine, first its virtual disk needs to be converted into a file format that Hyper-V can use. Metasploitable 2 is delivered as a VMware virtual machine, with a VMDK as its virtual disk file. The bad news is that Hyper-V can't understand how to parse VMDK files on its own, but the good news is that Microsoft provides a command-line utility that can convert the VMDK file into a VHDX file that Hyper-V *can* understand.

First, students will need to download and install the *Microsoft Virtual Machine Converter*. This application can be found at:

<https://www.microsoft.com/en-us/download/details.aspx?id=42497>

Note: Microsoft, like most software vendors, has a habit of reorganizing their knowledgebase and software distribution websites. That may result in this link being broken, or outdated. Additionally, the current version of the *Virtual Machine Converter* is 3.0. If there is a newer version available, be sure to download and use that instead.

This converter utility is a command-line application and specifically requires us to use Microsoft's PowerShell command-line interface. Depending on your system configuration, students may need to configure the PowerShell ExecutionPolicy option to allow us to load the converter module. To do this, press the Start button and in the search bar, search for powershell. Right-click the PowerShell icon, and select the option, *Run as administrator*. A UAC prompt will appear asking whether or not students want to allow PowerShell to make changes to their system. Click Yes to continue.

A window pops up, and the PowerShell prompt appears. Type in the command:

```
Set-ExecutionPolicy bypass
```

A notification labeled *Excution Policy Change* appears. This notification tells students that the execution policy protects users from powershell scripts they do not trust. Unfortunately, this policy needs to be changed to *bypass* in order for the converter to function properly. The notification asks, *Do you want to change the execution policy?* Type in 'y' and hit enter to confirm. The prompt returns indicating that the ExecutionPolicy has been changed.

Next, run the following command:

```
Import-Module 'C:\Program Files\Microsoft Virtual Machine Converter\MvmcCmdlet.psd1'
```

This command tells PowerShell "Hey, I want to load a PowerShell module (.psd1 file) from this location so that we can use its functionality."

Note: The file path for the `Import-Module` above is the default location that the *Microsoft Virtual Machine Converter* is installed to. If students installed the application to a custom path, adjust the `Import-Module` command accordingly.

If everything ran correctly you should get a new prompt back with no errors. Next, run the following command:

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath 'C:\Path\To\Metasploitable.vmdk'  
-VhdType DynamicHardDisk -VhdFormat vhdx  
-destination 'C:\Path\To\Virtual Machines\Metasploitable2'
```

Adjust the drive letters and directory paths as necessary for your Windows host. For example, let's assume students are using the default *Virtual Machines* path, created the `Metasploitable2` directory, then copied the `Metasploitable2-Linux` directory:

```
C:\ProgramData\Microsoft\Windows\Hyper-V\Metasploitable2\Metasploitable2-Linux
```

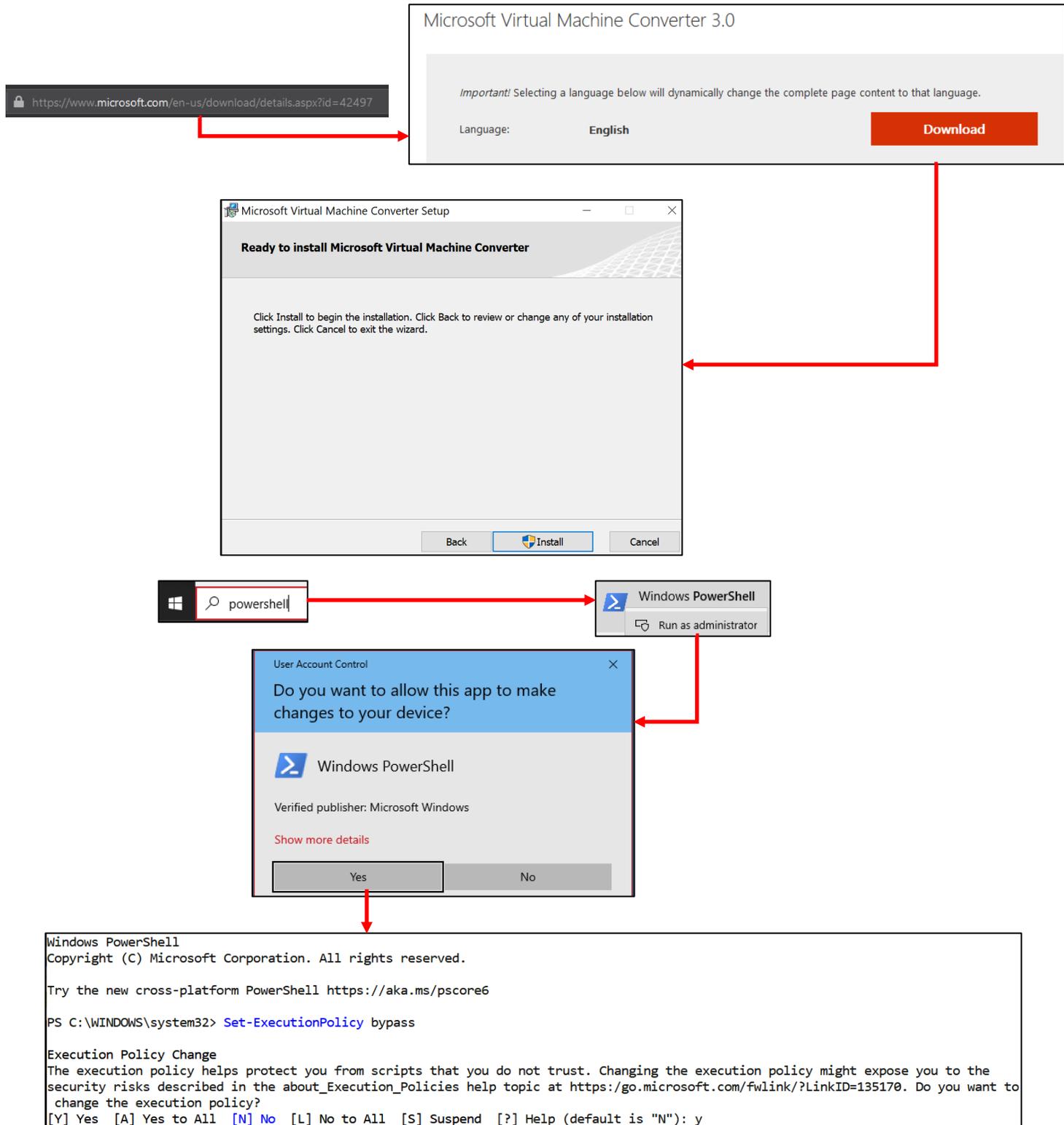
The command would look like this:

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath  
'C:\ProgramData\Microsoft\Windows\Hyper-V\Metasploitable2\Metasploitable2-  
Linux\Metasploitable.vmdk'  
-VhdType DynamicHardDisk -VhdFormat vhdx  
-destination 'C:\ProgramData\Microsoft\Windows\Hyper-V\Metasploitable2\'
```

On my system, the command looked like this:

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath  
'E:\VM-Configs\Metasploitable2\Metasploitable2-Linux\Metasploitable.vmdk'  
-VhdType DynamicHardDisk -VhdFormat vhdx  
-destination 'E:\VM-Configs\Metasploitable2\'
```

If the command was executed properly, the PowerShell prompt will change to reveal a progress bar as it converts the VMDK file to a VHDX file. See *fig. 9-64*, and *9-65* below.



9-64: Download and Install the *Microsoft Virtual Machine Converter*. The currently release is version 3.0. Note that the shield symbol next to the Install button indicates that attempting to install this software will trigger a UAC prompt. Next, students will need to open an elevated PowerShell session, and set the ExecutionPolicy to bypass.

```

PS C:\WINDOWS\system32> Import-Module 'C:\Program Files\Microsoft Virtual Machine Converter\MvmCmdlet.psd1'
PS C:\WINDOWS\system32> ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath 'E:\VM-Configs\Metasploitable2\Metasploitable2-Linux\Metasploitable.vmdk'
-VhdType DynamicHardDisk -VhdFormat vhdx -destination 'E:\VM-Configs\Metasploitable2\'
Converting drive E:\VM-Configs\Metasploitable2\Metasploitable2-Linux\Metasploitable.vmdk to dynamic VHDX.
Copying data from source to destination.
[oooo]
Destination                               Source
-----
E:\VM-Configs\Metasploitable2\Metasploitable.vhdx E:\VM-Configs\Metasploitable2\Metasploitable2-Linux\Metasploitable.vmdk

```

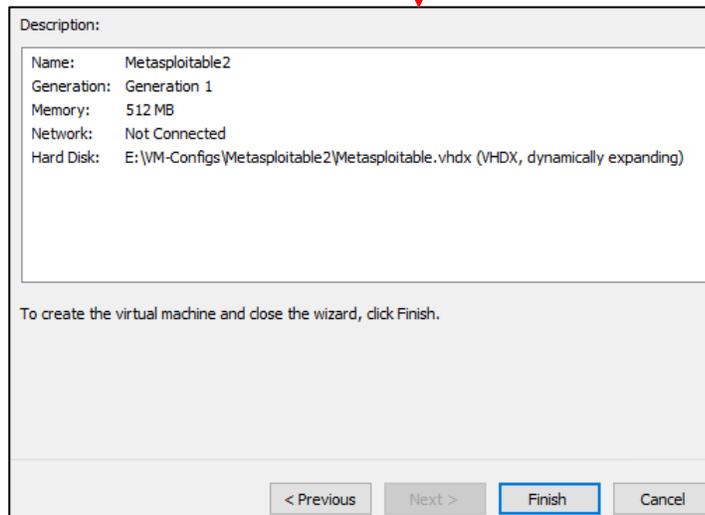
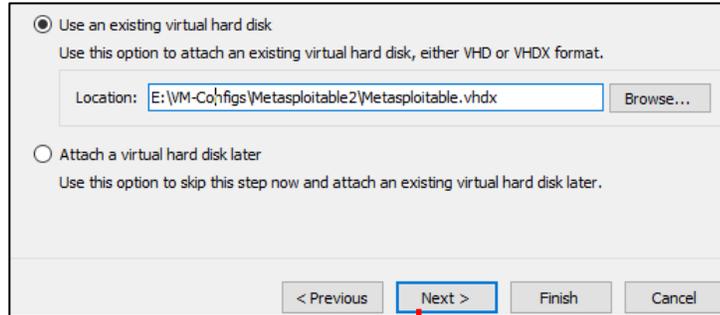
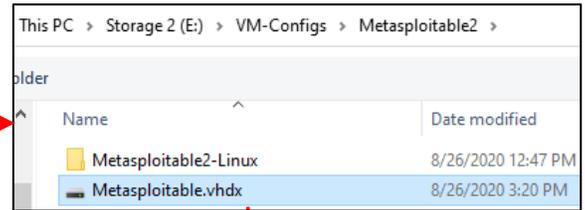
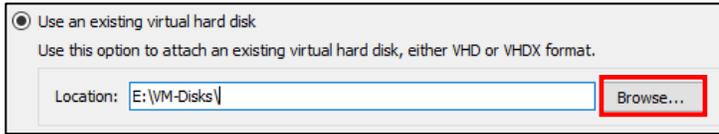
9-65: After setting the ExecutionPolicy, import the MvmcCmdlet.psd1 module, then run the ConvertTo-MvmcVirtualHardDisk command. The options '-SourceLiteralPath' and '-destination' will vary depending on where students store their virtual machines.

9.5.3.2 Creating the Metasploitable 2 VM

Open the *Hyper-V Manager* window, and begin the *New Virtual Machine* wizard. Create a Virtual Machine with the following settings:

Name:	Metasploitable2
Location:	Click the <i>Store the virtual machine in a different location</i> checkbox
	<Virtual Machines Directory>\Metasploitable2
Generation:	Generation 1
Memory:	Uncheck <i>Use Dynamic Memory</i>
	512MB
Networking:	Not Connected
Virtual Hard Disk Size:	We'll be getting to this in a moment...
Installation Options:	None! We'll be talking about this in a moment as well.

On the *Connect Virtual Hard Disk* screen, select the *Use an existing virtual hard disk* option, then click the *Browse* button to the right of the *Location* input box. Use the file browser and locate the *Metasploitable.vhdx* file created in the previous section, and select it. Once students click *Next*, the wizard displays the Summary screen. Click *Finish* to add Metasploitable 2 to the *Virtual Machines* listing.



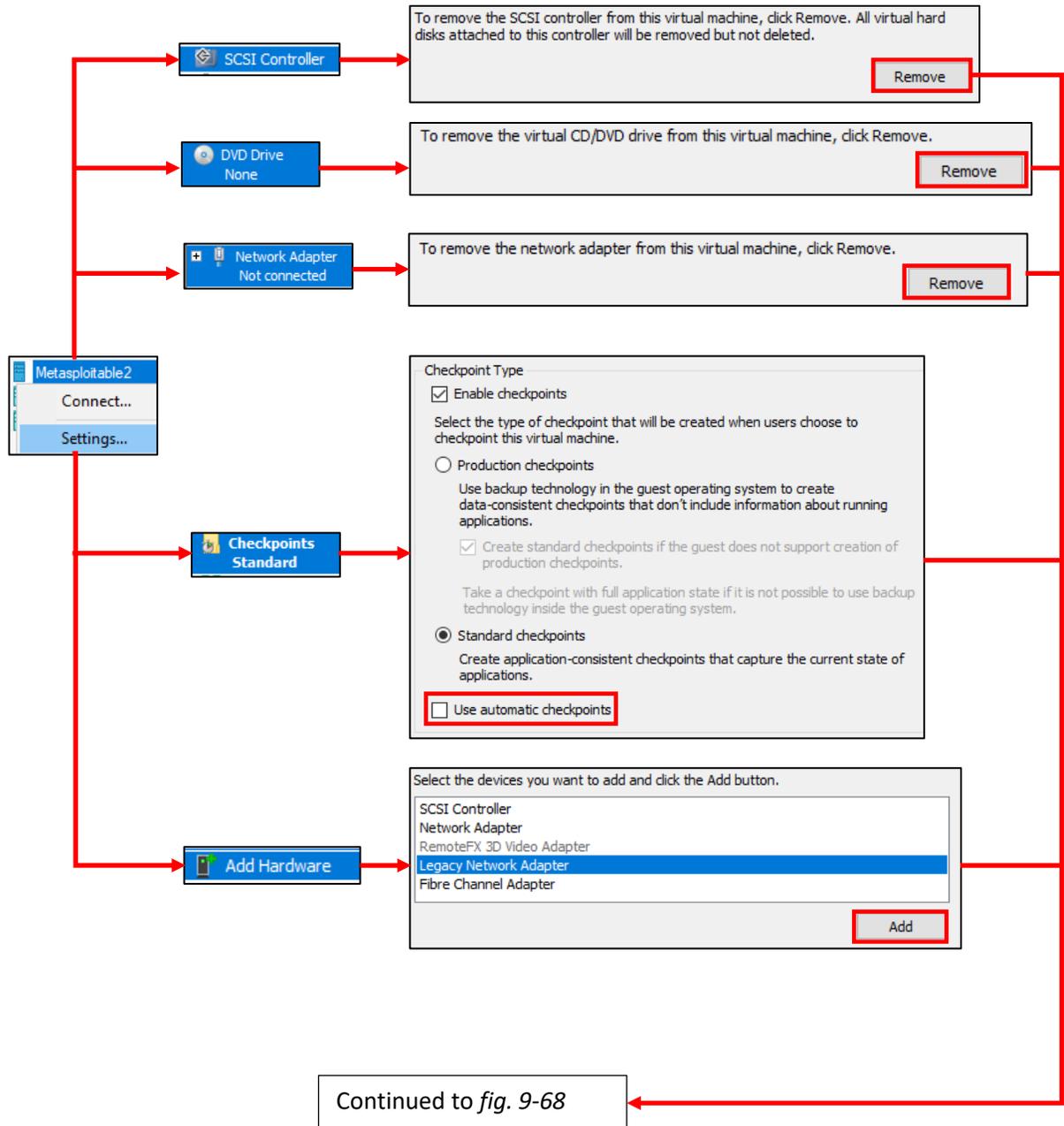
9-66: Start the *New Virtual Machine Wizard*, using the settings in the table on page 182. When students reach the *Connect Virtual Hard Disk* screen, click the *Use an existing virtual hard disk* radio button, and *Browse* to the *Metasploitable.vhdx* created in the previous section. Select it, proceed to the summary page, and *Finish* the wizard.

9.5.3.3 Adjusting Metasploitable 2 VM settings

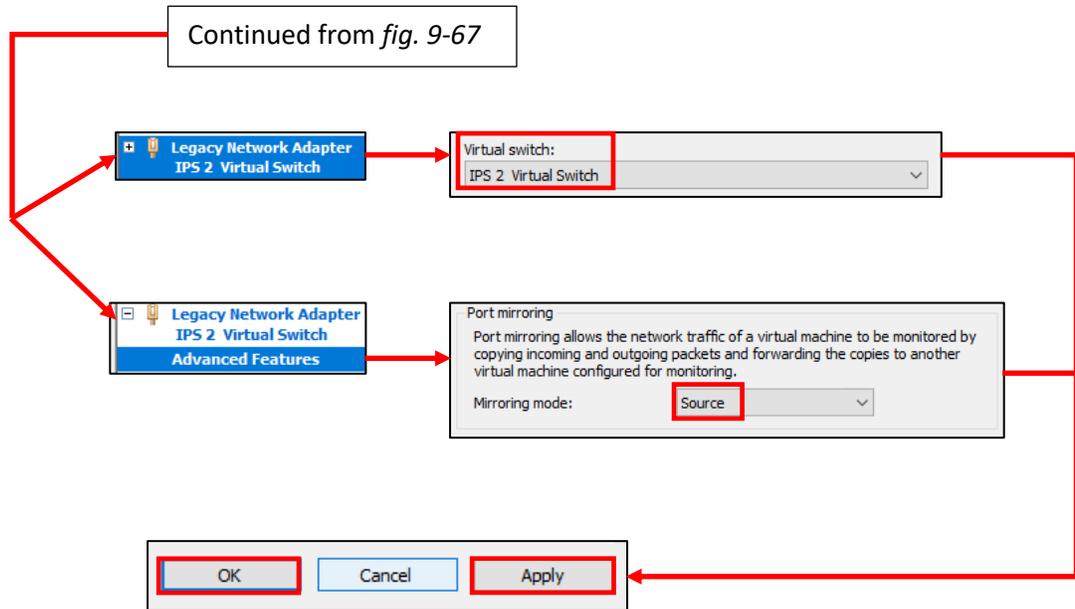
Before powering on the Metasploitable 2 VM, students will need to edit the virtual machine settings. The main difference between Metasploitable 2, and the other virtual machines is that students can perform most of the pre and post operating system install adjustments all at once, since Metasploitable 2 is a pre-built VM, with the operating system already installed. By this time, students should be comfortable with editing the settings of their virtual machines. Here is a list of configuration settings to change:

- *Remove the SCSI Controller*
- *Remove the DVD Drive*
- *Remove the Network Adapter*
- *Ensure Checkpoints are enabled, set to Standard checkpoints, and the Use automatic checkpoints option is disabled*

Afterwards, Click *Add Hardware*, highlight the *Legacy Network Adapter* option, and click *Add*. Highlight the *Legacy Network Adapter*, select the *IPS 2 Virtual Switch* from the *Virtual switch* drop-down. Click the + next to *Legacy Network Adapter*, then click on *Advanced Features*. As with the IPS, SIEM, and Kali VMs, we'll be coming back here to record the MAC address of this network interface later, but for now, scroll down to the *Port mirroring* section, and in the *Mirroring mode* drop-down, select *Source*. **Do not forget to perform this final task. It's extremely important!** Once finished, Click *Apply*, to confirm all of these configuration changes, then click *OK* to leave the *Settings* menu for the Metasploitable 2 VM.



9-67: Open up the *Settings* menu for the Metasploitable 2 VM. Remove the *Network Adapter*, *SCSI Controller* and *DVD Drive*, then navigate to *Checkpoints*, ensure checkpoints are enabled, *Standard checkpoints* are selected, and uncheck the *Use automatic checkpoints* checkbox. Select *Add Hardware*, select the *Legacy Network Adapter*, then click *Add*.



9-68: Highlight the newly created *Legacy Network Adapter*, and select the *IPS 2 Virtual Switch* option from the *Virtual Switch* drop-down. Click the + Symbol next to the *Legacy Network Adapter*, then select the *Advanced Features* option. Scroll down to the *Port mirroring* section and select *Source* from the *Mirror mode* drop-down. Once finished, click *Apply* then *OK* in the bottom right corner of the menu to apply these configuration changes, and exit the VM settings menu.

9.5.3.4 Booting Metasploitable 2

With metasploitable 2 fully configured, the next step is to power it on, and ensure that the virtual machine is functional. As always, right click on Metasploitable 2 in the Virtual Machines listing in the Hyper-V Manager, click *Start*, then right click on it again, and select *Connect*.

Wait for the metasploitable 2 to complete the boot up process, and eventually students will be greeted with a login banner. The banner itself informs users that the default credentials to log in are the username *msfadmin*, with the password *msfadmin*. Enter these credentials to log in. Upon successful login, type *exit* to log out, *Turn Off* the virtual machine from the virtual console navigation menu, then close the virtual console.

Right-click on the Metasploitable 2 VM in the *Hyper-V Manager* and select *Settings* again. Click the "+" sign next to the *Legacy Network Adapter*, then click *Advanced Features*. Under the *MAC address* section and the *Static* radio button, copy the MAC address that Hyper-V has generated, then log in to the pfSense webConfigurator. Navigate to *Services > DHCP Server > OPT1*, and create a new static DHCP mapping for the Metasploitable 2 VM, giving it the IP address 172.16.2.3. Save those changes, and that's it for now. The remaining four virtual machines have been successfully created.

```

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

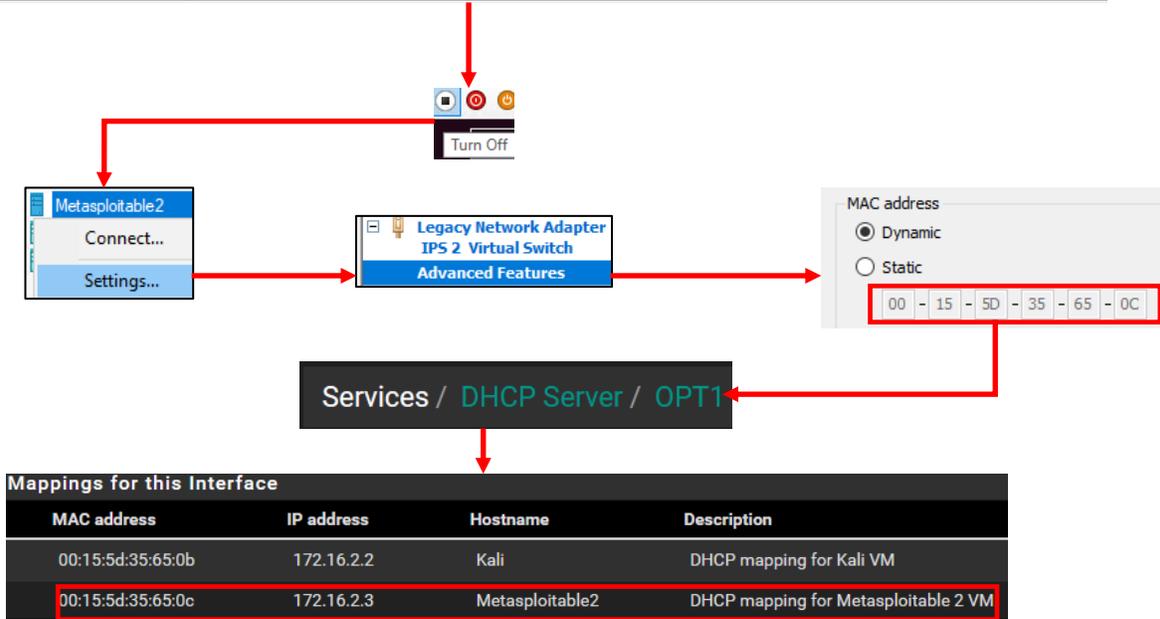
metasploitable login: msfadmin
Password:
Last login: Sun May 20 15:50:42 EDT 2012 from 172.16.123.1 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$

```



9-69: Start the Metasploitable 2 VM, Connect to its virtual console, confirm the login credentials work, then log out, and turn off the virtual machine. Students will not be doing any network diagnostics or connectivity checks, because the VM doesn't have an IP address or network connectivity right now. Check out the sidebar discussion *Why aren't we doing connectivity checks* below for a more in-depth explanation. Open up the Metasploitable 2 VM's Settings menu, and navigate to the *Advanced Features* option under the *Legacy Network Adapter*. Record the MAC address that Hyper-V assigned to this network interface, and use that to create a static DHCP mapping on the *OPT1* interface. Assign this MAC address the IP address 172.16.2.3.

Why aren't we doing connectivity checks?

Some of you may be wondering why we aren't doing connection checks or any of the stuff we did we for the SIEM, IPS, or Kali VMs, like checking the IP address or attempting to connect outbound. Well, that's because right now, the metasploitable 2 VM doesn't have an IP address at all. Don't worry, its intentional, and you'll be fixing this later. The reason metasploitable 2 doesn't have an IP address is that it's connected to the *IPS 2* network. While technically the *IPS 2* network shares the same subnet as *IPS 1*, and logically it's all a part of the *OPT1* network, *IPS 2* is its own physical network segment, and entirely separate from the *IPS 1* network. **Without something to bridge the *IPS 1* and *IPS 2* networks together, the *IPS 2* network is entirely isolated.**

Remember the network diagram back in [chapter 6](#) (p. 58)? The *IPS 2* network relies on the *IPS* virtual machine being fully configured and running either Snort3 or Suricata in AFPACKET bridging mode. No network bridge, no network connectivity. That means no IP address from the DHCP server, either. The static DHCP mapping we assigned for local network connectivity hasn't been assigned because the VM has no physical link to the DHCP server. You'll be fixing this later when you install either Snort3 or Suricata to the *IPS* virtual machine in chapter 17.

```
eth0    Link encap:Ethernet  HWaddr 00:15:5d:35:65:0c
        inet6 addr: fe80::215:5dff:fe35:650c/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:393 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:132822 (129.7 KB)
        Interrupt:9 Base address:0xec00
```

9-70: the output from `ifconfig -a` (without the `lo` interface output). `eth0` never got assigned an IPv4 address from DHCP because there is no physical connectivity between the *IPS 1* and *IPS 2* segments. We'll be solving this problem later when students install Snort3 or Suricata to the *IPS* virtual machine.

Okay, But Why the *Legacy Network Adapter*?

Some students may be wondering why the Metasploitable 2 VM is configured to utilize a *Legacy Network Adapter*. Well, it's because for one reason or another, Metasploitable 2 (Ubuntu 8.04) doesn't recognize the standard *Network Adapter* assigned to Hyper-V virtual machines by default. If I had to take a wild guess, it's probably because Ubuntu 8.04 doesn't have any drivers to recognize the standard network adapter. When I was initially troubleshooting this problem for the first book, I decided to try out the *Legacy Network Adapter* since Metasploitable 2 is so old, and it worked perfectly.

9.6 Checkpoints

The next (and final) task for students to perform will be creating baseline virtual machine checkpoints for the entire lab environment. Checkpoints (sometimes referred to as snapshots by other hypervisors) instruct the virtual machine's hypervisor to gather information about the VM's current state, and save it. Later on, if there is a problem with the virtual machine such as a malware infection, or a configuration problem that cannot be diagnosed, users can choose to restore the virtual machine to its state in the past, when the checkpoint was initially created.

Checkpoints can be created with virtual machines powered off, or while they are running, making them extremely versatile. Hyper-V virtual machines can also have more than one checkpoint, with the only limit being disk space required to hold them. It's extremely important to note that **virtual machine checkpoints are not a substitute for backups**. If students plan on running virtual machines with important data that they cannot afford to lose, checkpoints are not a substitute for backing up important files and data.

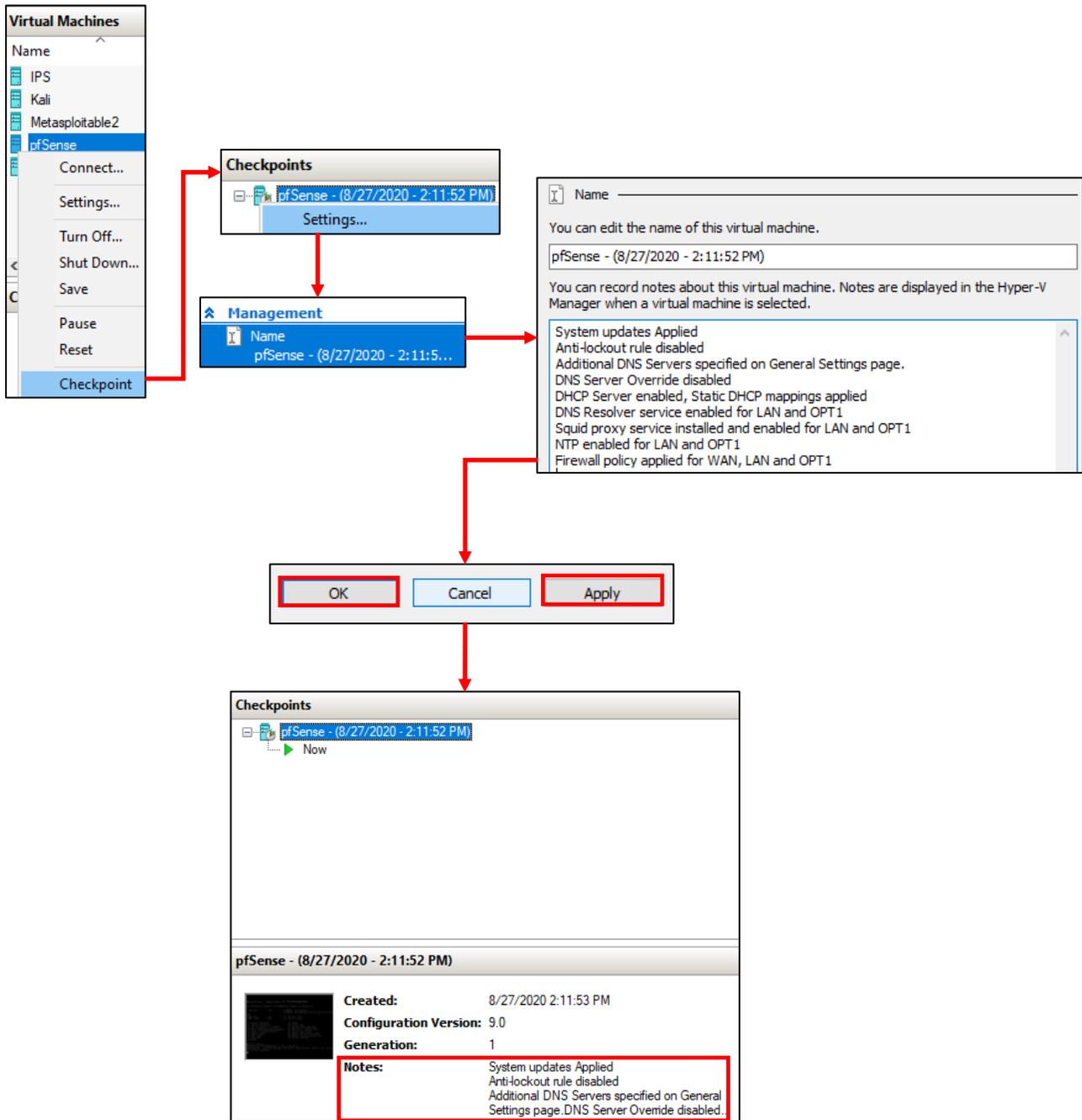
In this section, students will walk through the process of creating a virtual machine checkpoint for the pfSense VM. Afterwards, it will be left as an exercise to the students to repeat the process for the SIEM, IPS, kali, and Metasploitable 2 virtual machines. Once finished, students will be ready to move on with the configuration of their lab environment.

9.6.1 How to Create a Checkpoint

Open the *Hyper-V Manager* window, and right-click on the pfSense listing in the *Virtual Machines* section. Select the option labeled *Checkpoint*, and Hyper-V will instantly create a checkpoint of the pfSense VM in its current state. In the center pane, underneath the *Virtual Machines* window, is another window labeled *Checkpoints*. A new entry is listed, labeled pfSense – ([mm/dd/yyyy – timestamp]). Right click on that new entry, and select *Settings*. A window that is nearly identical to the virtual machine settings menu for pfSense will appear – in fact, this menu allows you to adjust the settings for the pfSense snapshot (as opposed to the pfSense VM in its current running state). Navigate to the menu option on the left labeled *Name*. The right pane updates with a pair of input boxes that students can edit to better describe the checkpoint they created. One of the fields can rename the checkpoint, while the other allows students to write notes. I would recommend editing the *Notes* field and describing the pfSense VM's condition. For example, here is what I documented in the notes input box:

System updates Applied
Anti-lockout rule disabled
Additional DNS Servers specified on General Settings page.
DNS Server Override disabled
DHCP Server enabled, Static DHCP mappings applied
DNS Resolver service enabled for LAN and OPT1
Squid proxy service installed and enabled for LAN and OPT1
NTP enabled for LAN and OPT1
Firewall policy applied for WAN, LAN and OPT1

Once finished, click the *Apply* button to commit these notes to the checkpoint, then click the *OK* button to close the checkpoint's *Settings* menu. In the Hyper-V Manager window, under the Checkpoints window, click the pfSense snapshot to highlight it, and a third window in the central pane appears, with a description of the snapshot. The Notes field updates to display some of the information students entered in the *Notes* field. Congrats, the pfSense VM now has its first snapshot, with proper documentation.



9-71: Right-click on the target VM and select the *Checkpoint* option. Hyper-V will instantly create a Checkpoint for the Virtual Machine. Highlight the newly created snapshot, right-click, open its *Settings* menu, and find the *Name* option. Put in some notes to describe the status of the VM, then click *Apply* and *OK* to apply the notes, then exit the *Settings* menu. Next, check the bottom window in the central pane of the *Hyper-V Manager* to confirm the notes are present.

9.6.2 Restoring a Checkpoint

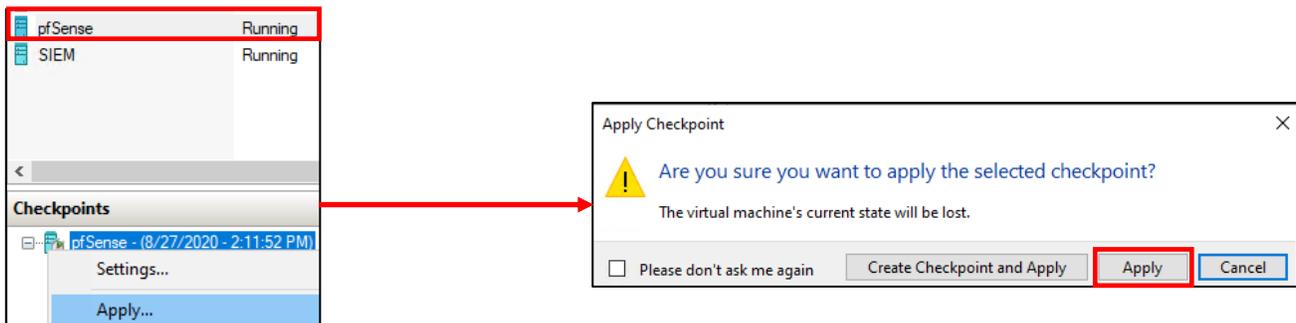
In order to restore a virtual machine checkpoint, select the target VM in the Hyper-V Manager to bring up its *Checkpoints* listing. Right click on the checkpoint students would like to restore, and click *Apply*. A pop-up window labeled *Apply Checkpoint* will ask students if they are sure they wish to apply the selected checkpoint, followed by three options:

Create Checkpoint and Apply: Creates a checkpoint of the VM in its current state before applying the requested checkpoint

Apply: Immediately applies the requested checkpoint

Cancel: Aborts the operation

Students may also click the Please don't ask me again checkbox, but this is one of the few times where it is not advised to do so. Click the *Apply* button to continue. Hyper-V will immediately restore the selected virtual machine's checkpoint.



9-72: To restore a checkpoint, highlight the target virtual machine to display its available checkpoints. Select the desired checkpoint, right click on it, select *Apply*, then click the *Apply* button again in the pop-up that appears. The checkpoint will be applied in moments.

9.6.3 Create checkpoints for the SIEM, IPS, Kali and Metasploitable 2 virtual machines

Now that students understand how to create checkpoints, it is highly recommended that they create baseline checkpoints for the SIEM, IPS, kali, and metasploitable 2 virtual machines. In the chapters to come, there will be a lot of complicated configuration tasks that students will need to perform in order to enable different functionality for their environment. Having a baseline snapshot to fall back to in case there are problems completing a task is handy for troubleshooting purposes.

Kali - (8/27/2020 - 7:52:20 PM)
You can record notes about this virtual machine in the Virtual Machine Manager when a virtual machine is selected.
OS installation completed static DHCP allocation applied network connectivity checks passed initial updates applied

Metasploitable2 - (8/27/2020 - 7:50:18 PM)
You can record notes about this virtual machine in the Virtual Machine Manager when a virtual machine is selected.
Successfully converted VMDK to VHDX Confirmed VM powers on Confirmed successful login No network connectivity (yet)

IPS - (8/27/2020 - 7:56:01 PM)
You can record notes about this virtual machine in the Virtual Machine Manager when a virtual machine is selected.
OS installation completed Static DHCP allocation applied Network connectivity checks passed Latest updates applied
IDS/IPS software not yet installed Splunk Forwarder not yet installed

SIEM - (8/27/2020 - 3:36:15 PM)
You can record notes about this virtual machine in the Virtual Machine Manager when a virtual machine is selected.
Initial OS install complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied
Splunk not yet installed

9-73: Now that students know how to create checkpoints, apply that knowledge and make baseline checkpoints for the other lab virtual machines. Having a baseline to fall back to in case something fails in the later chapters of this book is very important and will save students a lot of headaches.

Papers Plz?

When we created all of our Hyper-V virtual machines, I advised disabling the automatic checkpoints option. Why? Well, mainly because it uses a significant amount of disk space. I can't predict how much disk space students have to dedicate to their lab environment, and checkpoints can eat up a lot of space rapidly, so I chose to err on the side of caution and told everyone to disable them. The baseline checkpoints for all 5 VMs on my system totaled to about 4GB. For one checkpoint per VM.

If you happen to be running your lab environment on a system where disk space will never be an issue, or you want the extra piece of mind that automatic checkpoints bring you, then by all means, re-enable the feature. This your lab environment, and you can choose to run it however you see fit. However, even with automatic checkpoints enabled, if you are about to make a significant change to your virtual machines, I highly recommend creating a checkpoint manually, just as an extra bit of insurance.

9.7 Chapter Review

Students should have all 5 virtual machines created for the baseline lab environment, as well as baseline checkpoints for all 5 virtual machines. It was a long journey to get to this point, but it's far from over. Here is a checklist of tasks to complete:

- Complete [chapter 15, *Routing and Remote Access for Hosted Hypervisors*](#), starting on p. 727. In this chapter, students will learn how to enable SSH access to their lab virtual machines from Windows, Linux or MacOS. This functionality is vital for finishing the IPS and Splunk setup guides more easily than through the VM console alone.
- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in [chapter 17, *Network Intrusion Detection*](#), starting on p. 980.

Note: Throughout this chapter, several network interfaces across several virtual machines had the feature *Mirroring mode* (in the *Advanced Features* submenu) set to either *Source* or *Destination*. Make sure the following virtual machines have the following *Mirroring mode* configurations:

pfSense: The IPS 1 Virtual Switch network adapter should be set to *Source*. See [section 9.4.2, pp. 109-111](#).

Kali: the kali VM's only network adapter should be set to *Source*. See [section 9.5.1, pp. 135-140](#).

Metasploitable 2: set the Metasploitable 2 VM's *Legacy Network Adapter* to *Source*. See [section 9.5.3.3, pp. 184-186](#).

IPS: the network adapters attached to both the IPS 1 and IPS 2 virtual switches must be set to *Destination*. See [section 9.5.1, pp. 135-140](#).

Additionally, the network adapters attached to both the IPS 1 and IPS 2 virtual switches on the IPS virtual machine must have the *Enable MAC address spoofing* checkbox enabled in their respective *Advanced Features* sub-menu. See [section 9.5.1, pp. 135-140](#).

These configuration settings are absolutely vital to ensuring the IDS/IPS software functions correctly.

- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in [chapter 18, *Setting up Splunk*](#), starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out [chapter 19, *End of the Beginning*](#), starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out [chapter 20, *Extra Credit*](#), starting on p. 1055.

Chapter 10 Patch Notes

-Screen caps and diagrams all over this chapter, but unlike the first book, there aren't just one or two per page, figured out how to move and resize them all over the page. Hopefully, this will reduce the ginormous page count from the first book.

-The instructions on how to install VirtualBox are much more detailed. Instructions are available for Windows, MacOS, and Linux (installation via `.run` package)

-Explained to students that the properties menu for host-only network adapters is bugged for Linux and MacOS. Network configuration settings don't persist and don't apply.

-Provided detailed instructions on how to configure an IP address on the host-only interface on Windows, and the `vboxnet0` interface on both Linux and MacOS. Hinted that the network configuration will still not persist between reboots on Linux and MacOS, and recommend some scripts I developed to help work around this that I'll talk about much later, in chapter 15 (`flightcheck-linux.sh` and `flightcheck-osx.sh`)

-Offloaded instructions on how to install VM kernel extensions and decompressing metasploitable 2 and the pfSense ISO to chapter 1

-Sidebar discussion on the slight, but annoying differences in the VirtualBox VM settings menu on MacOS

-Talked about how the VirtualBox VM console will eat mouse/keyboard input and escape sequences to give the mouse/keyboard back to the host

-Detailed discussion on mapping VirtualBox network interfaces, to BSD interface names, to the pfSense firewall interface aliases, to the lab segments they are supposed to map to. It's very messy, very painful, and I had to design the illustrations to be printed in black and white.

-In the old book, each chapter had a webconfigurator initial setup section. It's all been moved to chapter 14. All the hypervisor guides redirect to chapter 14 after pfSense has IP addresses on all 3 interfaces. After completing chapter 14, readers are redirected back to their respective hypervisor setup guides.

-Put a large amount of effort into helping students troubleshoot network problems they may come up with when setting up pfSense as well as their other VMs. Troubleshooting situations like the WAN interface on pfSense (bridged to the local network) not getting an IP address, the local network using the same IP address range as the proposed network range for the virtual lab subnets, how troubleshoot the other virtual machines not getting their static DHCP allocations (comparing MAC addresses, etc.), what to do when network connectivity commands fail, etc.

Chapter 10: VirtualBox

VirtualBox is a hosted, (mostly) open-source hypervisor. Its greatest strengths are that it is free, runs on most modern operating systems, and is capable of importing virtual machines created by most other hypervisors seamlessly. As a direct result of the price being right, and all that flexibility, it has become ubiquitous – especially with regards to hands-on learning for most Computer Science, Information Technology, and/or Information Security classes. The first of many tasks readers will need to accomplish is acquiring VirtualBox and installing it, so let's get started. Open your web browser, and navigate to <https://www.virtualbox.org/wiki/Downloads>



VirtualBox

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0](#)

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2](#)

VirtualBox 6.1.8 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

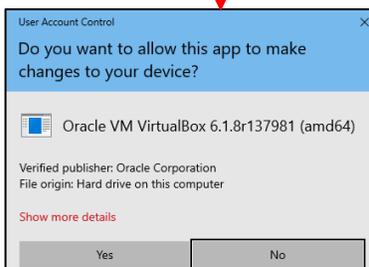
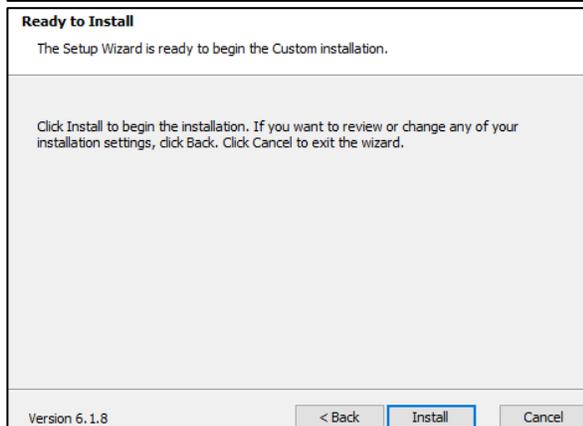
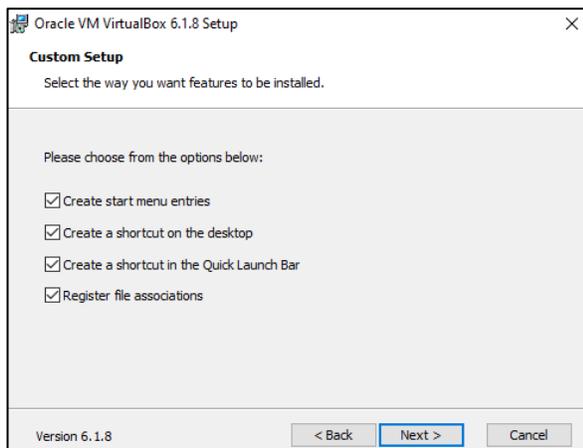
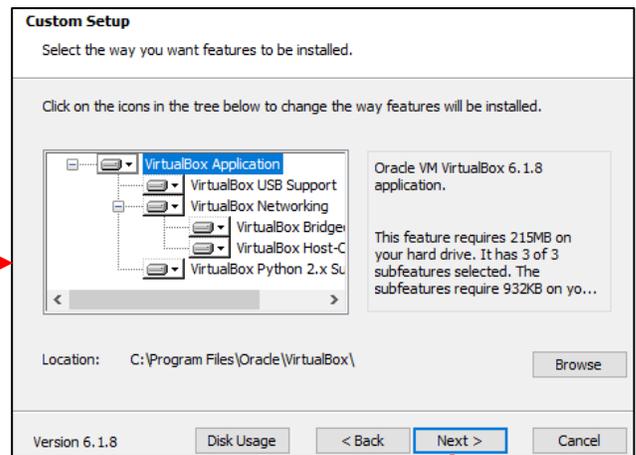
10-1: The VirtualBox download page. There are direct download links to the Windows, MacOS and Solaris installers. See section 10.3 for details on how to download VirtualBox for Linux.

10.1 Windows Installation Guide

In order to install VirtualBox, students will need Administrator rights on their system. Additionally, If UAC is enabled, users will need to grant the VirtualBox installer permissions to modify the system. After downloading the Windows installer (Virtualbox-[Current Version Number]-[Build Release]-Win.exe), double click it to start the installation wizard. Click *Next* to continue to the next screen of the wizard.

On the next screen, students may add or remove VirtualBox components, or change the installation directory. Click *Next* to accept the default settings. The screen that follows allows users to enable or disable the installer from creating shortcuts, and registering certain file extensions as being able to be opened by VirtualBox. Click *Next* to accept the defaults. The next screen has the text *Warning: Network Interfaces* in large, red text. This page states that the installer may temporarily disconnect the current system from the network in order to install network features for the hypervisors. Click *Yes* to continue. Students will be greeted with the text *Ready to Install*. If users want to change any installation settings, they can click the *Back* option. Otherwise, click *Install* to begin the installation process.

If UAC is enabled on Windows, a UAC prompt may appear, *Do you want to allow this app to make changes to your device?* followed by *Oracle VM Virtualbox[Version]r[BuildRelease] (amd64)*. Select *Yes* to allow the installer to continue. The final screen appears. Click *Finish* to exit the installer.



10-2: Windows installation process for Virtualbox. For the most part, the installation process is very straightforward. Students should be able to accept the default settings of the installation wizard and proceed through the installer. Please note if UAC is enabled, students may be required to accept some UAC prompts during installation.

Important Note for Apple M Series Hardware

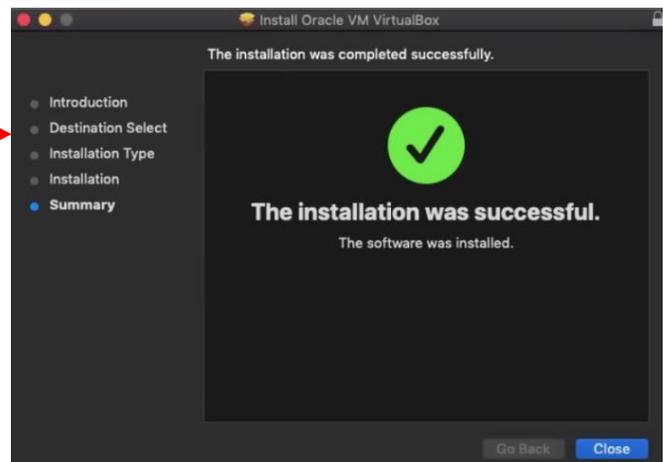
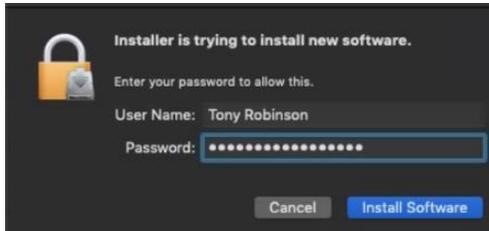
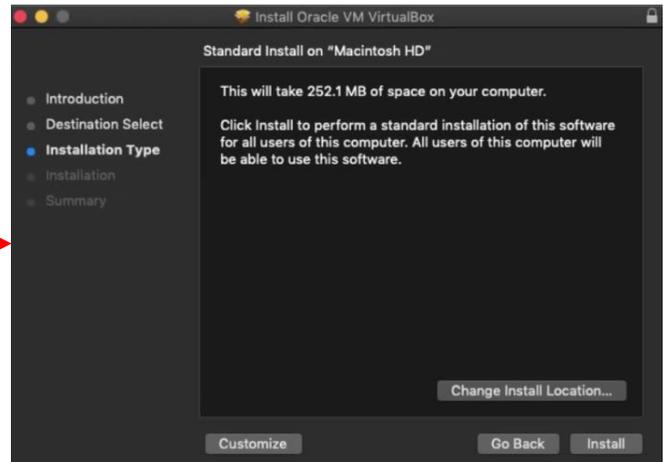
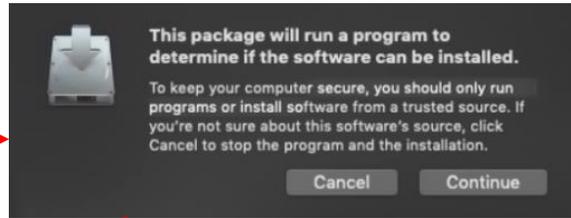
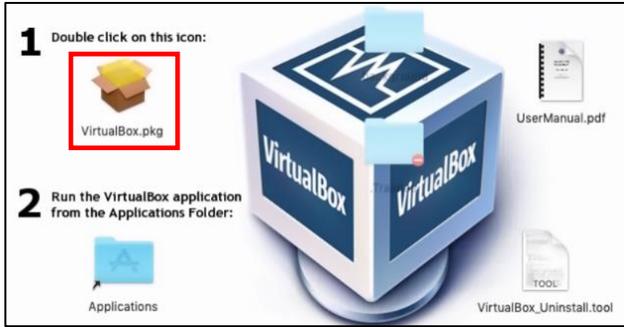
Recently, Apple has made a change to their product line that is having a huge impact on the state of virtualization on their platform. Apple is moving from Intel and x86 processors to their own in-house 'M' Series ARM-based processors. **This means that x86 hosted hypervisors like VirtualBox are not currently compatible with the new Apple M series hardware.** As of mine writing this (June 2021), there are no plans to bring VirtualBox to ARM processors.

10.2 MacOS Installation Guide

After downloading the installer file, (Virtualbox-[Version Number]-[Build Release]-OSX.dmg), locate the file in *Finder* (usually in the current user's Downloads folder) and double click it for MacOS to mount the image. After mounting the image, a small pop up window will appear. Double click the VirtualBox.pkg file to run the installer.

A small window will pop up that states "*This package will run a program to determine if the software can be installed.*" Click *Continue*. Provided everything goes well, students will be greeted by the VirtualBox installation wizard. Click *Continue* to move to the next screen and confirm where to install VirtualBox. The default installation location should be fine, so click *Install* to proceed. A new window titled "*Installer is trying to install new software*" appears. Enter your username and password, and click *Install Software* to let the VirtualBox installer continue.

After some time, the installer will confirm the installation was a success. Click *Close* to complete the installation. Afterwards, students may select *Keep* to hold on to the installer DMG, or *Move to Trash* to remove it. To run VirtualBox, students may need to open *Finder*, navigate to the Applications directory, and double click on the VirtualBox icon. I recommend pinning VirtualBox to the MacOS dock by right clicking the VirtualBox icon in dock bar (while the hypervisor is running), and selecting *Options > Keep in Dock*.



10-3: The MacOS Installation Process. Double-click the VirtualBox DMG package, then double click on the VirtualBox.pkg installer file. Much like the Windows installer, most of the installation process is straightforward. The VirtualBox installer will need the current user's password in order to finish the installation.

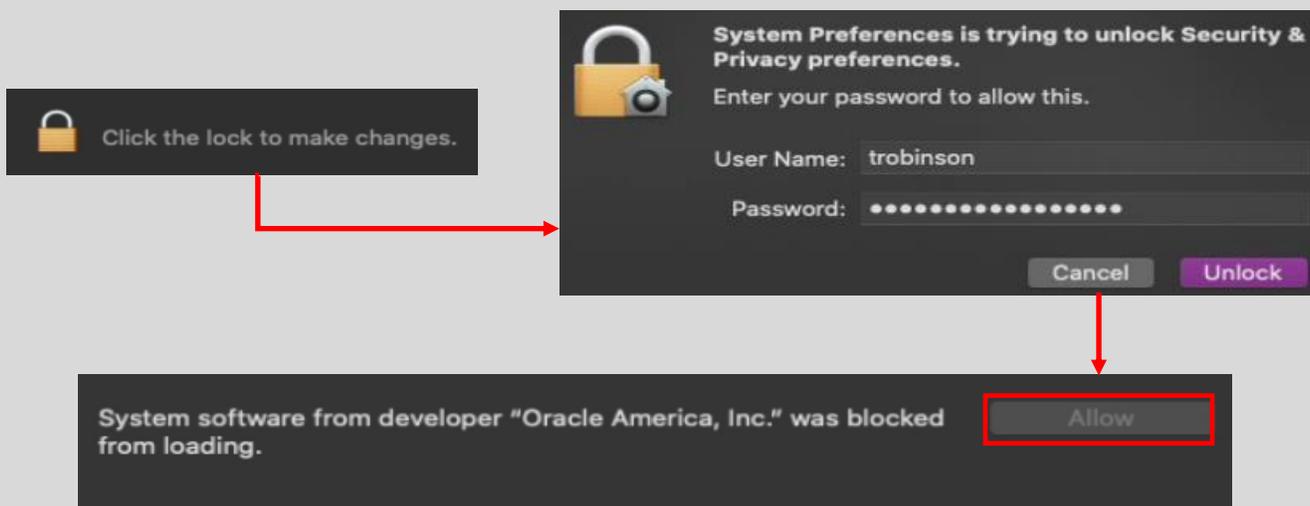
System Extension Blocked

Users might experience a problem in which MacOS issues a notification stating that the installer tried to install a system extension, and wasn't allowed to do so:



10-4: Ah, yes. Because I didn't already have to provide my password to install the software in the first place. Think different, not better.

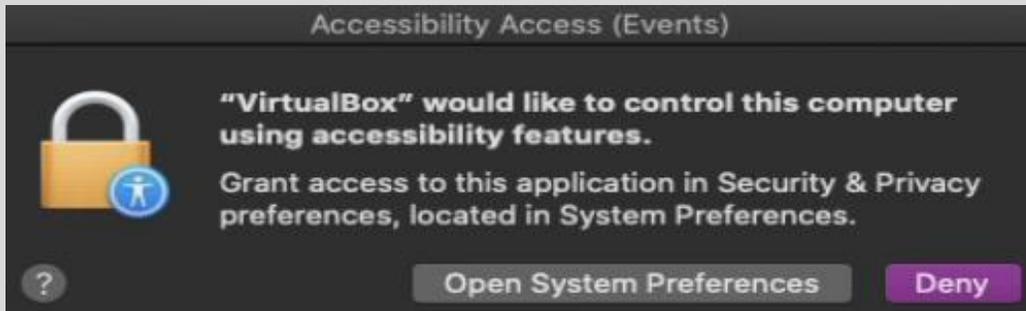
Click the *Open Security Preferences* button and MacOS will open the *Security and Privacy* settings window. Towards the bottom of the window, you'll see the message: *System software from developer "Oracle America, Inc." was blocked from loading*. You'll need to click the *Allow* button next to it. In order to do that (since its greyed out right now and you can't click it), find the padlock icon at the bottom of the window, and click on it in order to make system security changes. Enter your password in the dialogue box that appears, then click the *Allow* button (that should no longer be greyed out). If students are running MacOS Catalina, they may need to re-run the VirtualBox installer. For students running Big Sur or later, a reboot may be required.



10-5: After clicking *Open Security Preferences*, Click the padlock at the bottom of the *Security & Privacy Settings* window. Enter your password, then click the *Allow* button next to this ominous message. Students may be required to either re-run the Virtualbox installer, or reboot their system to use VirtualBox, depending on what version of MacOS they're running.

Mother, may I?

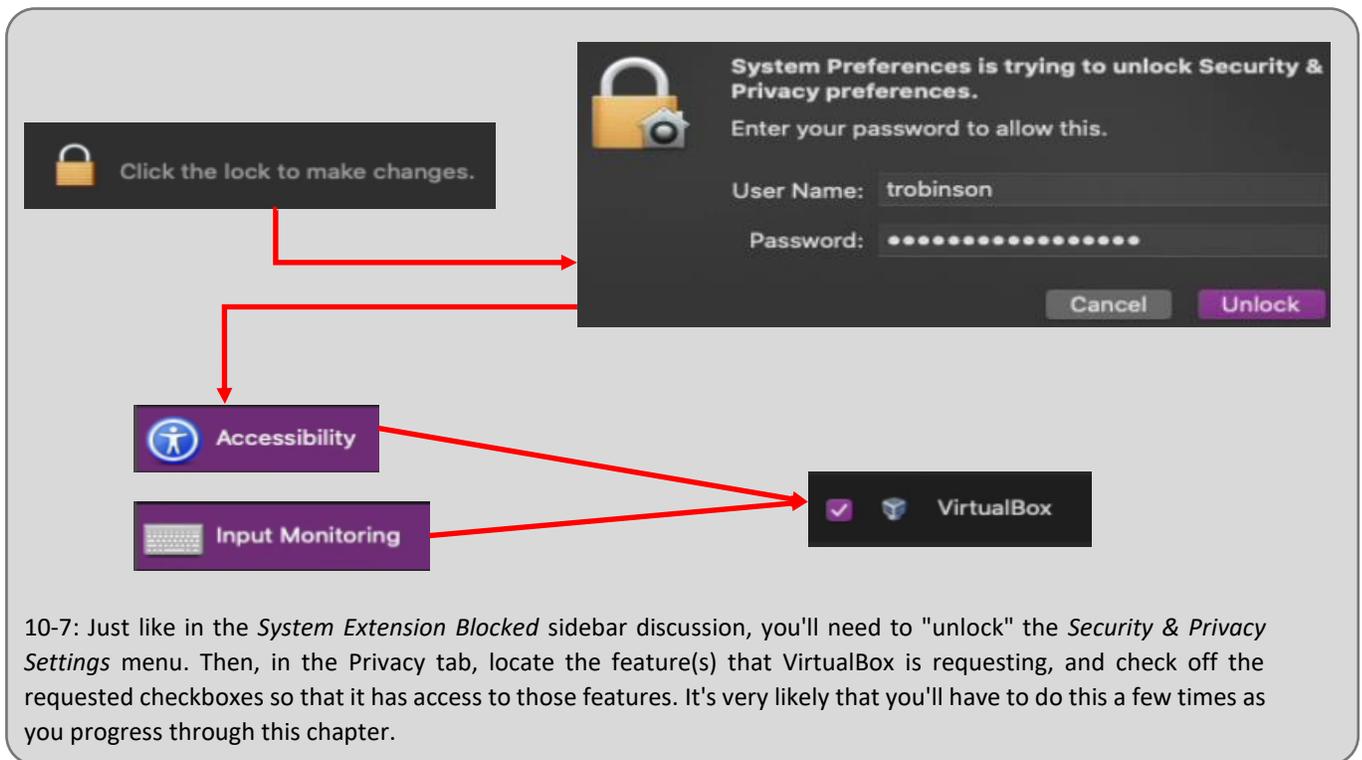
While following the instructions in this chapter, students running VirtualBox on MacOS may run into additional prompts in which the operating system will notify users that VirtualBox wants access to something, and then will either allow you to enable it on the fly, or unhelpfully tell you to go fix it yourself in the *Security & Privacy Settings* menu. Here is an example of what one of these pop-ups will look like when you see it:



10-6: You'll see pop-ups like this a lot as you continue to use VirtualBox on MacOS. The application installers don't supply a manifest of all the access they want, so you'll constantly be bombarded with notifications like this to fix the access violations yourself. Once in a while you can just click *Allow* and MacOS will fix it, but most of the time you're forced to find the permission the application wants and allow it yourself in the *Security & Privacy Settings* menu.

Most of the time when this happens, you'll need to click the *Open System Preferences* button. This will take you to the *Security & Privacy Settings* menu – specifically on the *Privacy* tab. Just like with the *System Extension Blocked* sidebar discussion above, you'll need to click the padlock and enter your password to make changes here. Once you've done so, select the setting that MacOS is complaining about (In this case, *Accessibility* settings), then click the checkbox next to VirtualBox to allow that specific setting.

Consider looking at some of the other options in the *Privacy* tab, to see if VirtualBox needs access to additional features while you're here. For example, I found another checkbox under the *Input Monitoring* setting that VirtualBox wants, but is unchecked. I never saw a dialogue box or a prompt to enable this access. I would advise enabling access to other features while you are here so that you're not bombarded with access requests, or interruptions while you're performing other tasks in this chapter.



10.3 Linux Installation Guide

The process for installing VirtualBox on Linux will be demonstrated using Ubuntu Desktop 20.04. However, the instructions should be easy to follow, regardless of the Linux distribution students prefer to use. The instructions provided assume readers already have either `root` or `sudo` access on their system to run the installer, and have already installed Linux kernel headers (see chapter 1, [section 1.7](#), pp. 28-32, for instructions on how to do this). Additionally, VirtualBox requires the software packages `make`, `gcc`, and `perl`. Students will need to acquire these software packages through their distribution's package manager.

There are two primary methods for installing VirtualBox on Linux – One of those methods is configuring your distribution's package manager to download and install it (and its prerequisites) for you, and the second method is through the use of a `.run` installer package. This installation guide will be using the `.run` method. While it requires students to do a little more legwork, the main advantage is that this installation method is compatible across a variety of Linux distributions.

Begin by visiting https://www.virtualbox.org/wiki/Linux_Downloads (click the *Linux distributions* link from the main download page), then click the link labeled, *All distributions*. This will download a package named `VirtualBox-[Version Number]-[Build Number]-Linux_amd64.run`. Please note that some browsers may require right clicking on the *All distributions* link and selecting *Save Target As* or *Save Link As* to begin downloading the file. On most Linux distributions, the default download location is the current user's Downloads directory (`/home/[username]/Downloads`, or `/root/Downloads` for the root user. `~/Downloads` can be used as a shortcut to access the Downloads directory, located in the current user's home directory).

Open a terminal window and navigate to the directory the `.run` package was downloaded to (e.g. `cd ~/Downloads`). The installer needs to be ran with root permissions. This can be done by running the command:

```
sudo bash VirtualBox-[Version Number]-[Build Number]-Linux_amd64.run
```

The name of the file is quite long, but you can use your command line shell's tab completion options to fill out the name of the file quickly. The command above runs the bash command interpreter with root access to execute the contents of the `.run` file. There are numerous other ways to run the installer, but this is the fastest and easiest method.

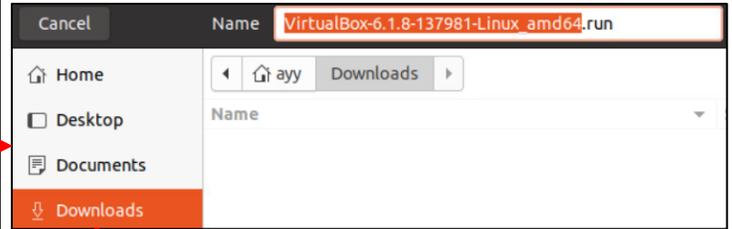
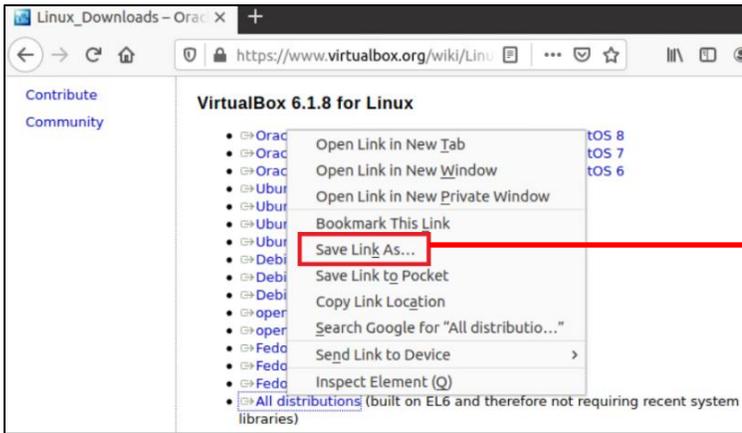
Once the installer is finished, the hypervisor's files will be installed to `/opt/VirtualBox`, while the `virtualbox` executable is located at `/usr/bin/virtualbox`. Most of the time, `/usr/bin` will be a part of the command shell's `PATH` variable. This means, `virtualbox` can usually be ran with the command:

```
virtualbox &
```

If you get the error `command not found`, try:

```
/usr/bin/virtualbox &
```

The ampersand (`&`) is a part of the command. It allows the user to keep using the terminal window while the `virtualbox` command is running. Be aware that if the terminal window used to run `virtualbox` is closed, the hypervisor will exit. If your distribution's window manager allows you to run `VirtualBox` directly, or allows users to create a shortcut on a dock of some sort, it is recommended to use these methods to run the hypervisor, in order to prevent it from shutting down accidentally.



```

ayy@ayy:~$ sudo apt-get update
[sudo] password for ayy:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
ayy@ayy:~$ sudo apt-get install make gcc perl
Reading package lists... Done
Building dependency tree
Reading state information... Done
perl is already the newest version (5.30.0-9build1).
Suggested packages:
  gcc-multilib autoconf automake libtool flex bison gcc-doc make-doc
The following NEW packages will be installed:
  gcc make
0 upgraded, 2 newly installed, 0 to remove and 55 not upgraded.
Need to get 0 B/168 kB of archives.
After this operation, 444 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y

ayy@ayy:~$ cd ~/Downloads
ayy@ayy:~/Downloads$ ls
VirtualBox-6.1.8-137981-Linux_amd64.run
ayy@ayy:~/Downloads$ sudo bash VirtualBox-6.1.8-137981-Linux_amd64.run
Verifying archive integrity... All good.
Uncompressing VirtualBox for Linux installation.....
VirtualBox Version 6.1.8 r137981 (2020-05-14T19:33:09Z) installer
VirtualBox has been installed successfully.

You will find useful information about using VirtualBox in the user manual
/opt/VirtualBox/UserManual.pdf
and in the user FAQ
http://www.virtualbox.org/wiki/User_FAQ

We hope that you enjoy using VirtualBox.

```

10-8: To Install VirtualBox on Linux, visit https://www.virtualbox.org/wiki/Linux_Downloads and download the .run package via the *All distributions* link. Students will need to install the make, perl, and gcc packages via their distribution's package manager before executing the .run file as the root user. VirtualBox will be installed to /opt/VirtualBox, and the virtualbox binary will be installed to /usr/bin/virtualbox.

Troubleshooting Installation Problems

In order to create this section for installing VirtualBox on Linux, I performed the `.run` installation method on Ubuntu Desktop 20.04 (Minimal Installation) and Redhat Linux 8 (Server with GUI). Here are a couple of problems I ran into, and recommendations on how to get around them:

- If EFI Secure Boot is enabled, The VirtualBox installer will fail stating that VirtualBox's kernel modules are not signed. There are two ways to get around this:
 - Boot into the EFI firmware (BIOS) of your system and disable Secure Boot (sometimes called "Trusted Boot", etc. depending on the motherboard and firmware manufacturer).
 - Install the `dkms` software package, using your distro's package manager. For Ubuntu users, this is pretty easy, but for Redhat users, you'll need to enable the EPEL software repository. In order to install it. You can learn about the EPEL software repository and how to enable it here: <https://access.redhat.com/solutions/3358>

- If students see the following error running the `virtualbox` command:

The `vboxdrv` kernel module is not loaded Either there is no module available for the current kernel (`kernel_version`) or it failed to load. Please recompile the kernel module and install it by

```
sudo /sbin/vboxconfig
```

You will not be able to start VMs until this problem is fixed.

- Begin by running: `sudo /sbin/vboxconfig`. Very likely, this command will fail with the error:
`vboxdrv.sh: failed: Look at /var/log/vbox-setup.log to find out what went wrong.`

- Read the output from `/var/log/vbox-setup.log`. Look for this line:

```
"Cannot generate ORC metadata for CONFIG_UNWINDER_ORC=y, please install libelf-dev, libelf-devel or elfutils-libelf-devel".
```

- I never experienced this error installing VirtualBox on Ubuntu. However, for Redhat Linux, I needed to install the software package `elfutils-libelf-devel`. After installing the `libelf` development package from your distro's package manager, re-run the command `sudo /sbin/vboxconfig`, then run the `virtualbox` command as normal.

Qt (BadWindow) errors, and how to fix them

If you are trying to start VirtualBox, and you get a series of errors that look something like this:

```
Qt WARNING: QXcbConnection: XCB error: 3 (BadWindow), sequence: ####, resource id:
#####, major code: ### (XXXXXXXXXXXX), Minor code: ##
```

Open a terminal window and run the following commands:

```
export QT_SCREEN_SCALE_FACTORS=1
export QT_SCALE_FACTOR=1
export QT_AUTO_SCREEN_SCALE_FACTOR=0
```

Then run `virtualbox &` (or `/usr/bin/virtualbox &`). If the VirtualBox Manager window appears, problem solved! These commands set a series of environment variables for Qt, a core component of the Gnome Desktop Window Manager. These environment variables unfortunately will not be applied permanently. As soon as the system is rebooted (or the terminal window is closed) the environment variables will no longer be set. Here is a group of commands you can run to make these configuration settings permanent, if they helped fixed your problems:

```
cp ~/.bashrc ~/.bashrc_backup
echo "export QT_SCREEN_SCALE_FACTORS=1" >> ~/.bashrc
echo "export QT_SCALE_FACTOR=1" >> ~/.bashrc
echo "export QT_AUTO_SCREEN_SCALE_FACTOR=0" >> ~/.bashrc
```

First, we use the `cp` command to make a copy of the `.bashrc` file in the user's home directory in case there is a problem and the file gets deleted or otherwise corrupted. In a nutshell, the `.bashrc` file is used to set individual user preferences when they log in, or start terminal sessions. The `echo` commands will append the three `export` commands above to the end of the current user's `.bashrc` file, ensuring that these commands are run, and the environment variables are set every time you want to run VirtualBox.

```
Qt WARNING: QXcbConnection: XCB error: 3 (BadWindow), sequence: 488, resource id
: 46137356, major code: 2 (ChangeWindowAttributes), minor code: 0 1
```

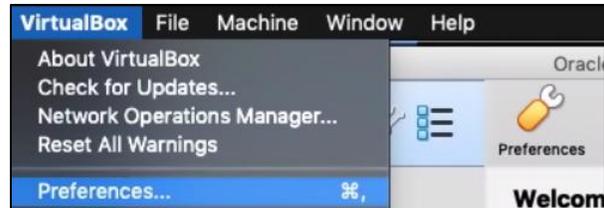
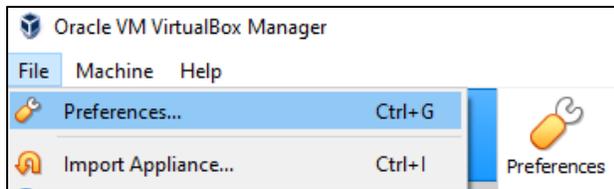
```
export QT_SCREEN_SCALE_FACTORS=1
export QT_SCALE_FACTOR=1
export QT_AUTO_SCREEN_SCALE_FACTOR=0 2
```

```
cp ~/.bashrc ~/.bashrc_backup
echo "export QT_SCREEN_SCALE_FACTORS=1" >> ~/.bashrc
echo "export QT_SCALE_FACTOR=1" >> ~/.bashrc
echo "export QT_AUTO_SCREEN_SCALE_FACTOR=0" >> ~/.bashrc 3
```

10-9: Are you getting a ton of "Qt Warning" messages (1) when attempting to start VirtualBox? Open up a terminal window and run the `export` commands (2) listed above, then try running the `virtualbox` command again. If successful, you'll want to run the `cp` and `echo` commands listed above (3) to add those `export` commands to your user's `bashrc` file. This will make it so where those specific environment variables and settings VirtualBox needs in order to display properly are automatically set when you log in.

10.4 Customizing VirtualBox

When starting VirtualBox for the first time, users are greeted with the VirtualBox Manager's welcome screen. VirtualBox has a host of customization options available that can be accessed through the *Preferences* menu. This menu can be reached by clicking the *Preferences* button (with the wrench icon) on the Virtual Manager window, or through your operating systems' navigation menu. For Linux and Windows users, select *File > Preferences...* While MacOS users should select *VirtualBox > Preferences*.



10-9: Configuration options for VirtualBox can be accessed through the *Preferences* icon on the VirtualBox Manager screen, regardless of the host operating system you are using. Windows and Linux users may also access the *Preferences...* option through the navigation menu under *File*, while MacOS users may select *Preferences* under the *VirtualBox* option on their navigation menu.

The *Preferences* menu has a host of configuration options available. They can be accessed by clicking the icon associated with the setting users wish to modify. Many of the configuration options are pretty self-explanatory. Here are some of the more important options available to students:

The *Network* option may be useful for users who may be required to create a NAT network for working around network access controls on their physical network. This situation was discussed in-depth in Chapter 4, [section 4.1.2, NAT Networking \(and Port Forwarding\)](#), pages 46-47.

The *Extensions* option allows users to install plugins and extensions to VirtualBox in order to extend the functionality of the hypervisor itself. For example, VirtualBox has an extension package that users may install here to enable USB 2.0 and 3.0 support for virtual machines.

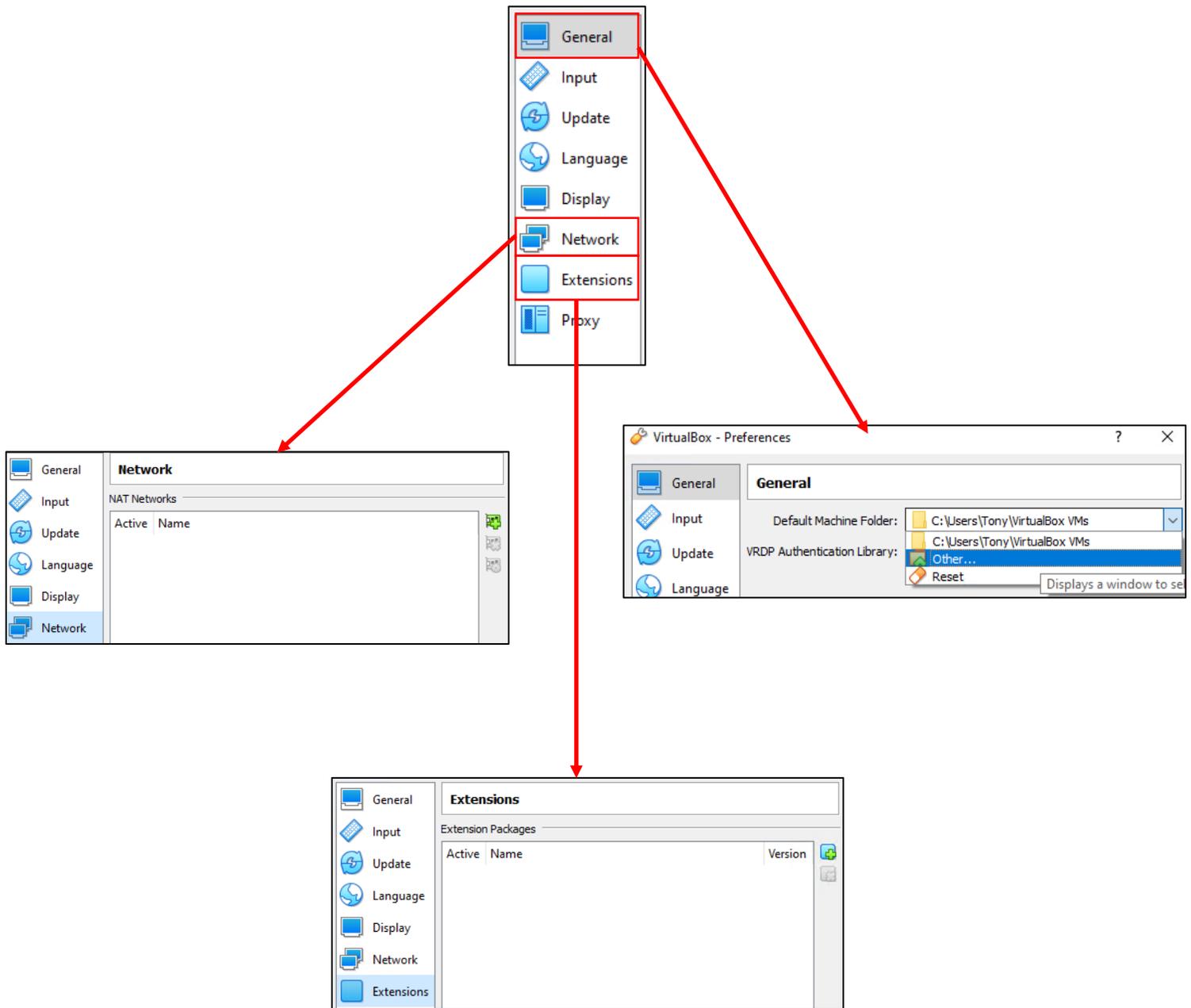
The *General* option allows users to define the VRDP authentication library (which we will not be touching), or define an option called the *Default Machine Folder*. By default, VirtualBox places virtual machine files in subfolders under the folder named "VirtualBox VMs". This folder is located in the current user's home directory.

For Windows users this is usually: C:\Users\[username]\VirtualBox VMs

For Linux users: /home/[username]/VirtualBox VMs

Finally, for MacOS users: /Users/[username]/Virtualbox VMs

Students may wish to change the *Default Machine Folder* to another directory on a separate, dedicated disk drive for better I/O performance.



10-10: The VirtualBox *Preferences* menu. Depicted are the *Network*, *Extensions* and *General* settings. Most of the configuration are self-explanatory, and more often than not, the default settings will be acceptable. The only setting that may need adjustment is the *Default Machine Folder* under *General*. This configuration option defines where VirtualBox will store virtual machine files by default. If students have a dedicated disk set up for their virtual machines, this setting may need to be modified to use that disk.

10.5 Configuring the Host-Only Virtual Network Adapter

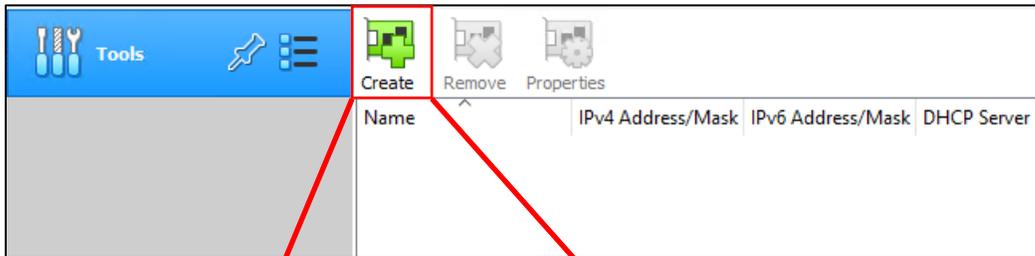
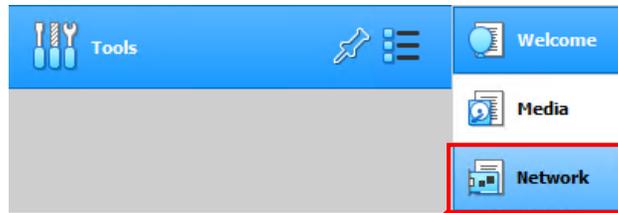
In this section, students will explore the *Host Network Manager* (which is not to be confused with the *Network* option under the *Preferences* menu) to create a host-only network interface for their host system, and disable the VirtualBox DHCP server. In some cases, a host-only interface may have already been created by default. If that is the case, students are still advised to follow along to ensure that the DHCP server has been disabled.

In the main *VirtualBox Manager* window, along the left portion of the window is a box labeled *Tools*. Towards the center of the screen is an icon that looks like a bulleted list – three blue square bullets, next to three black lines. Clicking this icon reveals a small menu with the options *Welcome*, *Media*, and *Network*. Click the *Network* option to open the *Host Network Manager*. Alternatively, users may select *File > Host Network Manager...* from the navigation menu, to reach the same destination.

The *Host Network Manager* has three main features: Creating host-only network adapters, removing them, or adjusting their properties. These features are accessed through the *Create*, *Remove* and *Properties* buttons, respectively. In some cases, VirtualBox may have already created a host-only network adapter as a part of the installation process. If so, the network adapter will be listed in the window below under the *Name* column. On Windows, the interface is named *VirtualBox Host-Only Ethernet Adapter*, while on Linux and MacOS, the name of the interface is *vboxnet0*. If you do not see a network interface listed under the *Name* column, click the *Create* icon to create a host-only network adapter. Windows users should note that if UAC is enabled, a prompt will pop up. They will need to click *Yes* to allow VirtualBox to create the new interface. **The lab environment will only require one host-only network interface. Make sure there is only one interface listed in the Host Network Manager.**

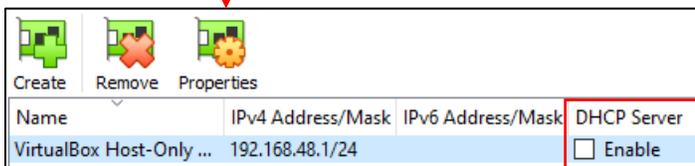
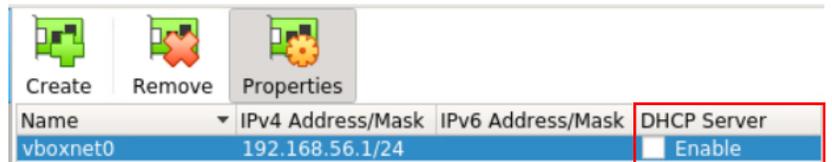
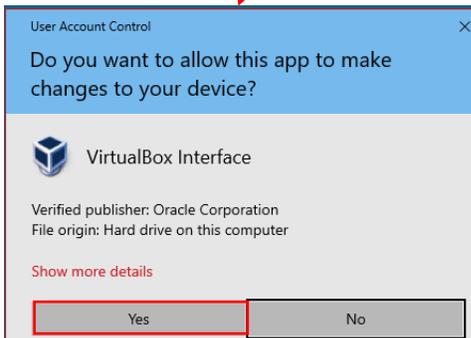
Another one of the information columns on the Host Network Manager screen is labeled *DHCP Server*, and contains a single checkbox. If the box is checked, that means the VirtualBox DHCP service is enabled. If it is unchecked, that means it is disabled – the state we want it to remain in.

Notice that the *IPv4 Address/Mask* column is already filled out. VirtualBox supplies a default IP address and subnet mask to its host-only interfaces. Do not pay attention to this, we'll be using operating system tools to change the IP address of this interface in just a moment.



Windows

Linux/MacOS



10-11: This illustration describes the process to create a host-only network adapter on both Windows and Linux/MacOS. Windows users may be required to click through a UAC prompt. Make sure the checkbox labeled *Enable* under the DHCP Server column is unchecked. Ensure there is only one host-only network interface – vboxnet0 (Linux/MacOS) or VirtualBox Host-Only Ethernet Adapter (Windows).

Why aren't we using the *Properties* button?

More adventurous students may have also noticed that by highlighting a host-only interface and clicking the *Properties* button, they can access tabs to configure network adapter itself, or the DHCP server for the network associated to that host-only adapter. Well, we aren't using VirtualBox's DHCP server, because the pfSense DHCP server is better and easier to manage in practically every way. So, if students unchecked the DHCP Server checkbox, they have no reason to bother with *DHCP Server* tab.

As for the *Adapter* tab, many would be led to believe that this tab is a much easier way to assign an IP address to host-only network adapters, without having to futz around with network configuration tools like `ifconfig`, `ip addr`, or `ncpa.cpl`, right? Well, in normal circumstances, you would be correct, but I've experienced buggy behavior where IP address settings configured in the *Adapter* tab did not "stick" or properly apply after changing them and hitting the *Apply* button. So, I would personally advise against trusting it, or using it to manage the IP addresses of any host-only network adapters until Oracle fixes it, or removes it.

10.5.1 Setting the Host-Only Adapter's IP Address

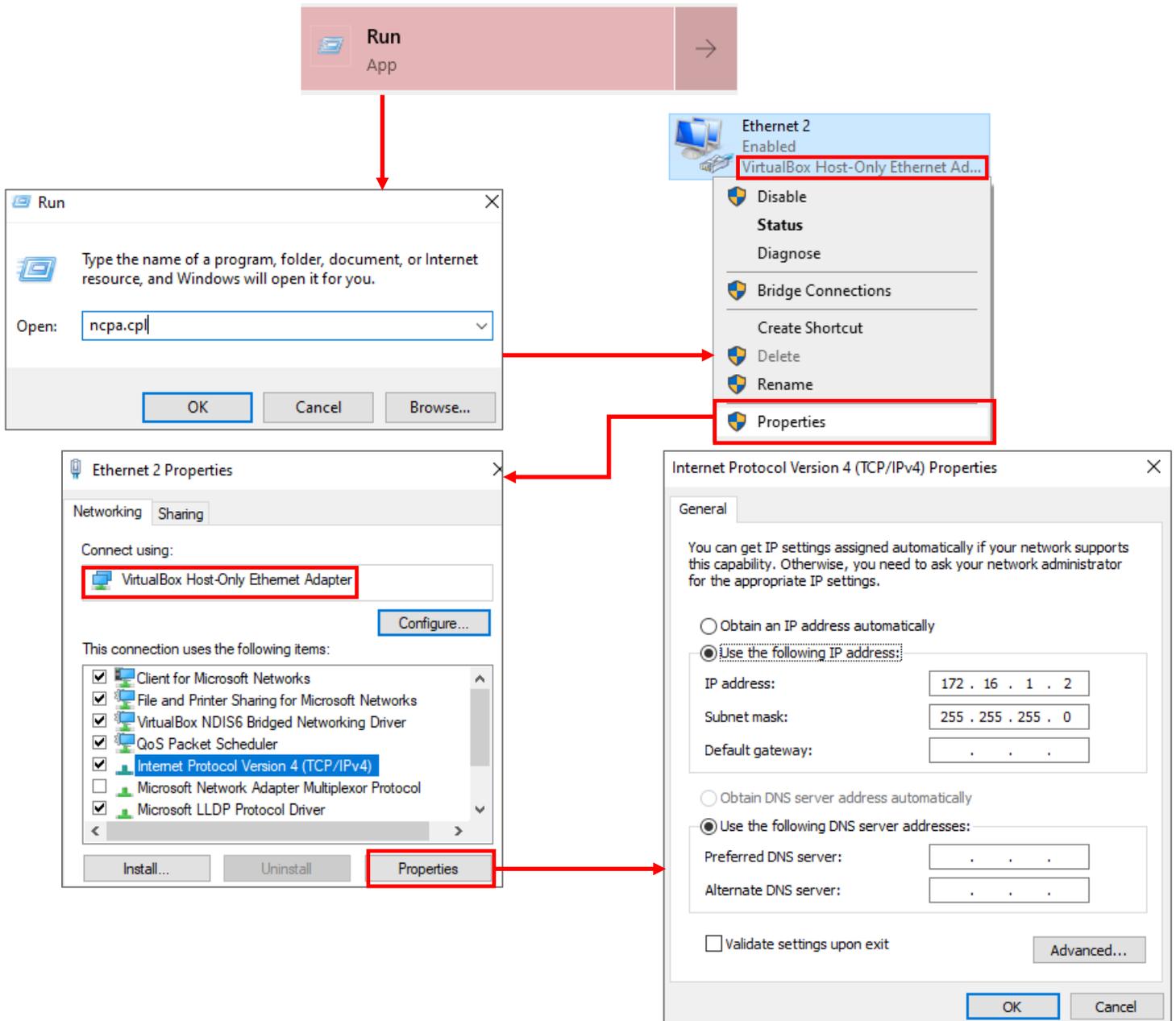
Every major operating system has some method of manually configuring the IP address of a connected network adapter – in many cases, more than one method. This section will show you how to use the network control panel on Windows (`ncpa.cpl`), `ifconfig` on MacOS, and `ip addr` on Linux to set the IP address and subnet mask of your host-only network adapter.

10.5.1.1 Windows and `ncpa.cpl`

Begin by opening the Start menu, and searching for 'run', to locate the run prompt application (alternatively, you can use the Windows/Meta + R key to instantly open the run prompt). Enter `ncpa.cpl` into the *Open* input box, and hit enter. This will open up the *Network Connections* panel without the need to figure out where Microsoft hid it in the latest seasonal update.

Locate the network adapter icon that has the light gray text *VirtualBox Host-Only Ethernet Adapter* (On my computer, this was a network adapter labeled *Ethernet 2*, but your mileage may vary), right-click on that icon, and select *Properties*. In the box labeled *This connection uses the following items*, locate the entry labeled *Internet Protocol Version 4 (TCP/IPv4)*, left-click to highlight it, then click the *Properties* button.

This opens a new window labeled *Internet Protocol Version 4 (TCP/IPv4) Properties*. In the *General* tab below, make sure that the *Use the following IP address* radio button is selected, as well as the *Use the following DNS server addresses* radio button. In the *IP address* input box enter `172.16.1.2`, and in the *Subnet mask* input box, enter `255.255.255.0`. **All of the other remaining input boxes should be left blank.** Click *OK* to apply these settings.



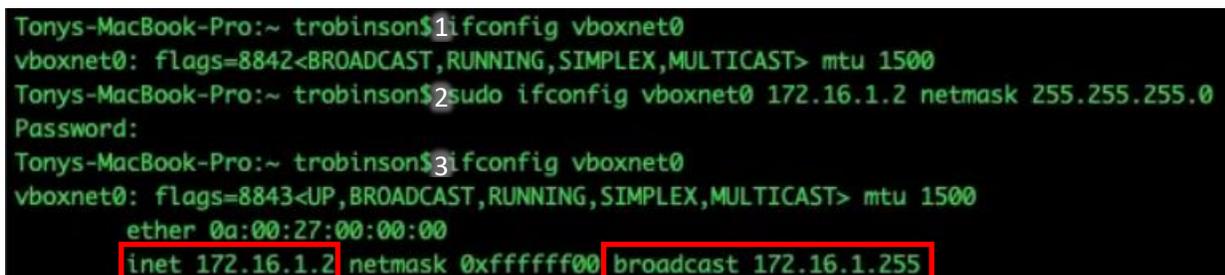
10-12: Access the *Network Connections* panel (ncpa.cp1). Once there, find the VirtualBox Host-Only Ethernet Adapter, and modify the properties of Internet Protocol Version 4 (TCP/IPv4) to configure the IP address. **Do not set a default gateway or any DNS server addresses.**

10.5.1.2 MacOS and ifconfig

Open up the *iTerm* or *iTerm2* terminal application, and enter the commands:

```
ifconfig vboxnet0
sudo ifconfig vboxnet0 172.16.1.2 netmask 255.255.255.0
ifconfig vboxnet0
```

The first `ifconfig` command is to confirm that the `vboxnet0` interface exists. The second command uses `sudo` to gain root permissions in order to use the `ifconfig` command to manually configure an IP address and subnet mask for `vboxnet0`. Finally, the third command (exactly the same as the first) is used to verify that the correct IP address and subnet mask has been applied. The `inet` field should contain the value `172.16.1.2`, and `172.16.1.255` for the `broadcast` field.



```
Tonys-MacBook-Pro:~ trobinson$1ifconfig vboxnet0
vboxnet0: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
Tonys-MacBook-Pro:~ trobinson$2sudo ifconfig vboxnet0 172.16.1.2 netmask 255.255.255.0
Password:
Tonys-MacBook-Pro:~ trobinson$3ifconfig vboxnet0
vboxnet0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 0a:00:27:00:00:00
inet 172.16.1.2 netmask 0xfffff00 broadcast 172.16.1.255
```

10-13: `ifconfig` is ran the first time (1) to make sure `vboxnet0` is present on the system, the second time (2) with `sudo` in order to change the IP address and subnet mask of `vboxnet0`, and the third time (3) to confirm the IP address and netmask has been applied correctly via the `inet` and `broadcast` fields.

10.5.1.3 Linux and ip addr

Some Linux distributions still ship with the `ifconfig` command, or have it available as some sort of legacy network tools package. If it is available and/or installed on your host system, you may follow the same instructions in section 10.5.1.2 above. However, most distros have switched to use the `ip` command suite for accessing network information and modifying the status of network interfaces on the command-line.

Open a terminal application and run the following commands:

```
ip addr show dev vboxnet0
sudo ip -4 addr flush label "vboxnet0"
sudo ip addr add 172.16.1.2/24 dev vboxnet0
ip addr show dev vboxnet0
```

`ip addr` is ran four times. The first time is to verify `vboxnet0` exists. The second time, with `sudo`, is in order to remove any default IP address configurations assigned by VirtualBox. The third time, also with `sudo`, is in order to set the IP address and subnet mask for `vboxnet0`. The fourth and final time is to confirm the IP address and subnet was properly configured. The `inet` field should contain the value `172.16.1.2/24`.

```

ayy@ayy:~$1ip addr show dev vboxnet0
3: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state
  DOWN group default qlen 1000
    link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
ayy@ayy:~$2sudo ip -4 addr flush label "vboxnet0"
[sudo] password for ayy:
ayy@ayy:~$3sudo ip addr add 172.16.1.2/24 dev vboxnet0
ayy@ayy:~$ ip addr show dev vboxnet0
3: vboxnet0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state
  DOWN group default qlen 1000
    link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.2/24 scope global vboxnet0
      valid_lft forever preferred_lft forever

```

10-14: The first `ip` command (1) is used to make sure `vboxnet0` exists. The second `ip` command (2), ran with `sudo` privileges, is used remove any IP address configurations VirtualBox may have applied to `vboxnet0` when it was created. The third command (3) is to set the IP address (172.16.1.2) and subnet mask (/24 – another way to display 255.255.255.0 called slash notation) for use with the lab environment. Finally, the fourth `ip` command is used to confirm the IP address and subnet mask has been successfully applied. The `inet` field should contain the value 172.16.1.2/24.

Linux/MacOS Students: `vboxnet0` and its IP address settings have disappeared. Why?

Linux and MacOS users may have noticed that upon rebooting their host operating system, `vboxnet0` and any interface configurations performed (like, say setting the IP address and subnet mask) are gone. Starting the VirtualBox Manager will at least bring back `vboxnet0`, but the IP address configuration is still gone. Unfortunately, this is something of a known issue with some hypervisors on MacOS and/or Linux – **this is not a VirtualBox-only problem.**

Unfortunately, after having done this for years, I am still unaware of any method of getting the `vboxnet0` interface itself, and/or any customizations performed against the interface to persist between reboots on Linux or MacOS. The only method that seems to work would be to restart VirtualBox, and reperform the steps necessary to set the IP address and subnet mask for `vboxnet0` on your host OS as necessary.

Fortunately, I've created a simple script available for both Linux and MacOS users that automates this process. We'll talk more about this script in Chapter 15. For now, I ask readers to bear in mind that every time the host system reboots, they will be required to restart their hypervisor and reconfigure the IP address of `vboxnet0` manually.

10.6 Building the first Virtual Machine, pfSense

The pfSense virtual machine is responsible for binding the entire lab environment together. It is a well-supported firewall distribution with amazing ease of use and functionality. pfSense is also very modular, featuring a system for adding on additional functionality through BSD's pkg software package manager.

As a reminder, it is recommended for students to download all of the ISOs, and pre-built virtual machines required for their lab environment in advance. Check out chapter 1, [section 1.5.4](#) (p. 26) for download links. Additionally, students must decompress the pfSense installation ISO before attempting to boot from it. [Section 1.8](#) (pp. 33-35) covers how to do this on Windows, Linux or MacOS.

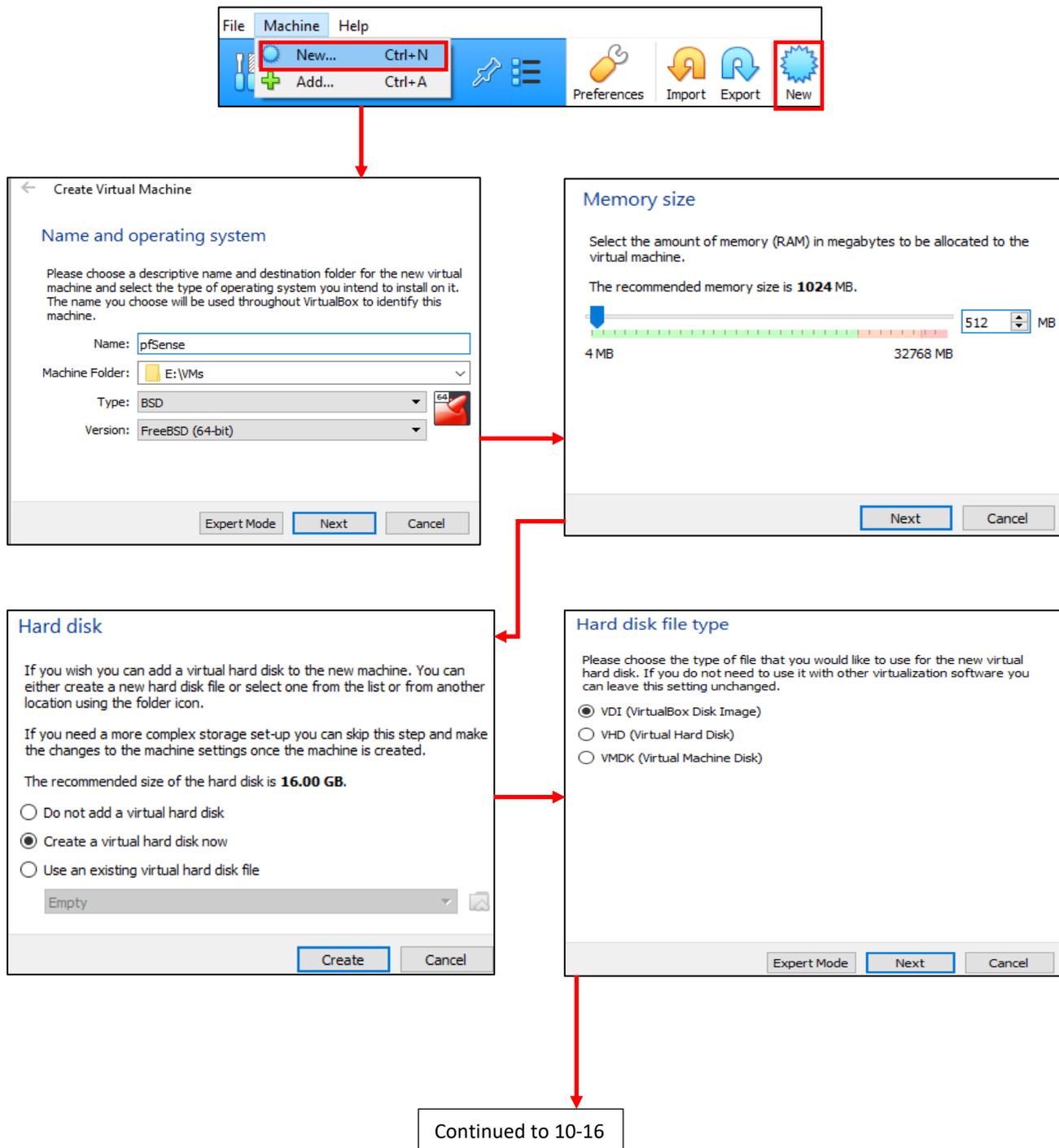
10.6.1 VM Creation

VirtualBox VMs are created through a step-by-step process using the *Create Virtual Machine Wizard*. Click the *New* icon in the *VirtualBox Manager* window, or click *Machine > New...* in the navigation menu to begin.

The first screen, titled *Name and operating system* has the *Name* input box for users to provide the new VM with a name, the *Machine Folder* setting to configure where the virtual machine's files will be stored upon creation, and last but not least, the *Type* and *Version* drop-down boxes. The *Type* and *Version* settings tell VirtualBox what guest operating system will be used in this VM so that it can customize settings and virtual hardware for better compatibility. Name the VM pfSense and use the default machine folder location. For the *Type*, select BSD, and for the *Version*, select FreeBSD (64-bit). Afterwards, click *Next* to continue.

The next screen is titled *Memory size*. Using the slider or the input box, adjust the memory down to 512MB, then click *Next*.

The *Hard disk* screen appears. By default, the radio button labeled *Create a virtual hard disk now* should be selected. If not, select it then click the *Create* button. You'll then be prompted to choose a *Hard disk file type*. The default should be *VDI (VirtualBox Disk Image)*. If not, select it then click *Next*. The *Storage on physical hard disk* screen provides the options to either dynamically allocate disk space as the virtual machine needs it, or set the virtual hard disk to a fixed size. Select the *Fixed size* radio button, then click *Next* to continue. The next screen, *File location and size* asks to confirm where the VDI file the wizard will create should be stored, and its size. The default file location in the input box should be fine. Use the slider and/or input box next to the slider to reduce the size of the file to 5.00GB, then click *Create*. Depending on the performance of the drive chosen to store the VDI file, this may take a bit of time. Upon completion, the new VM shows up in the VirtualBox Manager window.



10-15: Start the *Create Virtual Machine Wizard* by clicking either the *New* icon in the *VirtualBox Manager* window, or selectin *Machine > New* From the navigation menu. Name the virtual machine *pfSense*, accept the default *Machine Folder* location, Set the *Type* as *BSD*, and the *Version* as *FreeBSD (64-bit)*. Set the *Memory size* to *512MB*. Select *Create a virtual hard disk now*, then select *VDI (VirtualBox Disk Image)* as the *Hard disk file type*.

Continued from 10-15

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

Dynamically allocated
 Fixed size

Next Cancel

File location and size

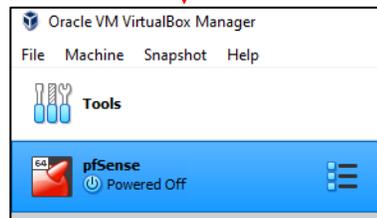
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

E:\VMs\pfSense\pfSense.vdi

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4.00 MB 2.00 TB 5.00 GB

Create Cancel



10-16: Select *Fixed size* for *Storage on Physical hard disk*. On the *File location and size* screen, accept the default location for the `pfSense.vdi` file, set its size to 5.00GB, then click the *Create* button to finish the wizard. When complete, a new entry labeled `pfSense` should appear in the *VirtualBox Manager* window.

10.6.2 pfSense Virtual Machine Settings (Part 1)

With the virtual machine created, we're just about ready to install an operating system. Before proceeding, there are a handful of adjustments to make to the virtual machine, and its hardware. These adjustments are to ensure the VM does not have any extra functionality enabled that is not needed. This makes it a little bit leaner, and reduces possible attack surface.

The pfSense VM appears immediately beneath the *Tools* option in the *VirtualBox Manager* window. This area of the window is the virtual machine inventory – it's the collection of all VMs the VirtualBox hypervisor is currently aware of. The inventory shows that our new VM is currently powered off. To the right of the name and power status of the virtual machine is the same bulleted list icon observed earlier next to the *Tools* option. Clicking on it opens a small menu with the options *Details*, *Snapshots*, or *Logs*. The *Details* option will change the right side of the window to display more detailed information about the virtual machine, such as resource allocations, and currently assigned virtual hardware. The *Logs* option opens a console that can be used to review log files pertaining to the operation of the currently selected virtual machine. The *Snapshots* setting allows users to review snapshots for the currently selected VM. Snapshots will be covered later.

Students will need to access the virtual machine settings window. As with most of the functions so far, there are numerous ways to do this. The easiest method involves right-clicking the pfSense VM in the inventory listing, and selecting the *Settings* option with the small, golden gear next to it. Alternatively, with the pfSense VM highlighted, students may click the large gear icon near the top of the VirtualBox Manager window, labeled *Settings*. A new window opens labeled *[VM name] – Settings*. The left side of this window has a menu with a wide variety of choices available.

Students start in the virtual machine's *General* settings. This menu has four tabs labeled *Basic*, *Advanced*, *Description*, and *Disk Encryption*. Many of the options in the various tabs are pretty self-explanatory. We're interested in the *Advanced* tab, specifically, the drop-down menus labeled *Shared Clipboard*, and *Drag'n'Drop*. Make sure both the *Shared Clipboard* and *Drag'n'Drop* drop-downs are set to *Disabled*.

Next, click on *System* to open up the *System* menu. This menu has three tabs labeled *Motherboard*, *Processor*, and *Acceleration*. The *Motherboard* tab is our destination, and its settings should be displayed by default. Pay attention to the window labeled *Boot Order*, students will be back here to change this later. For now, confirm the drop-down labeled *Pointing Device* is set to *PS/2 Mouse*.

Navigate to the *Storage* menu option next. This menu is divided into two columns labeled *Storage Devices* and *Attributes*. Under *Storage Devices* is a small, blue disc, representing the virtual CD/DVD ROM drive for the virtual machine. Clicking on the disc icon highlights it, and changes the contents of the *Attributes* column. To the left of the drop-down menu labeled *Optical Drive*, is another, smaller disc icon. Clicking on it reveals a drop-down menu. Select the option

Choose a disk file... and a file browser window will open. Browse to the location of the decompressed pfSense ISO file and select it. The *Information* section below will update with information about the ISO selected, and the disc icon under *Storage Devices* will update from *Empty* to the name of the ISO file selected (e.g. pfSense-CE-2.4.5-RELEASE-amd64.iso). **This is required for us to install the pfSense operating system later in this chapter.**

The *Audio* options menu is next. Uncheck the checkbox labeled *Enable Audio*. None of the VMs in the lab environment will require audio support. If done correctly, the audio options will all be greyed out.

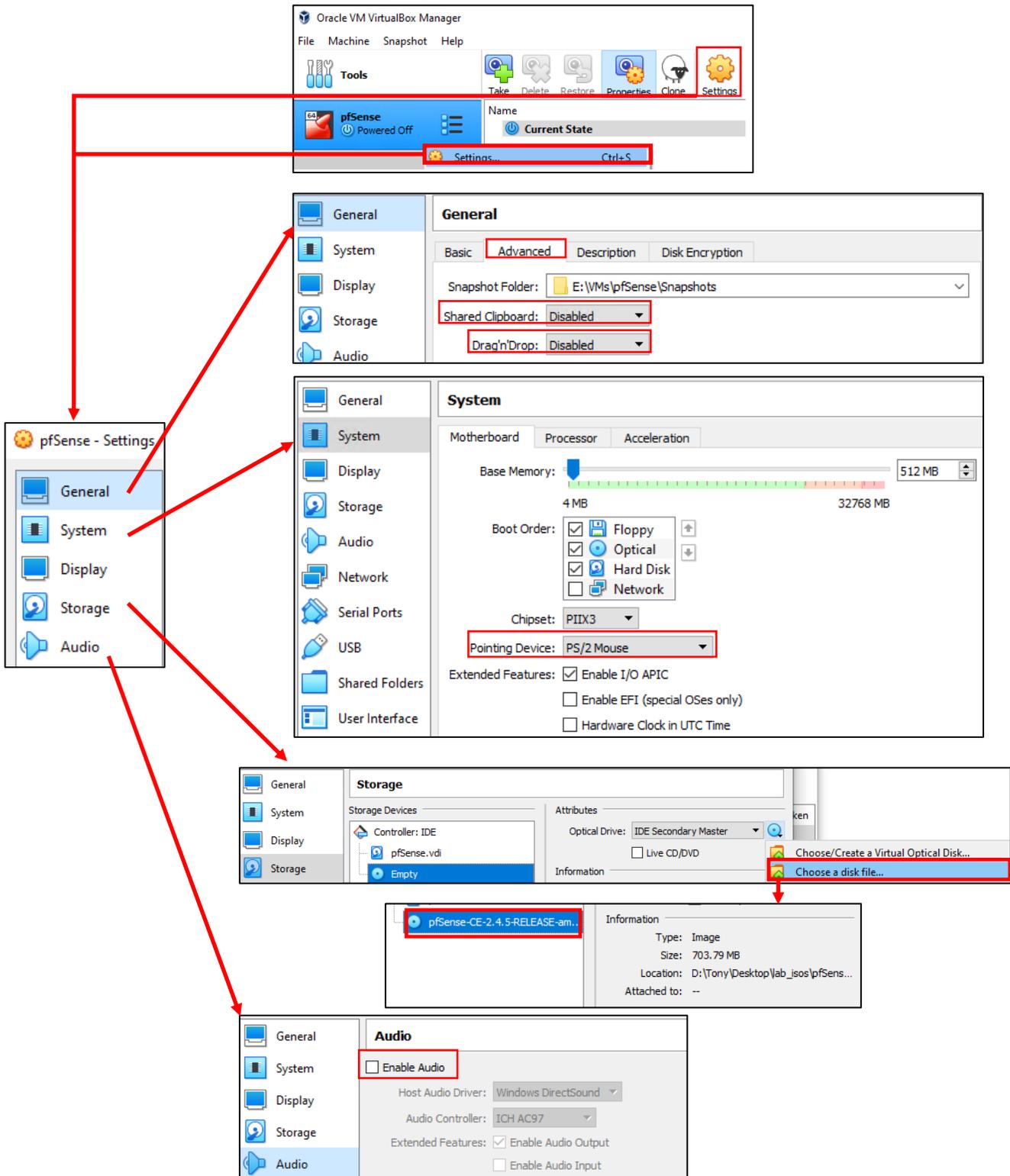
Skip over the *Network* option for now, and select *Serial Ports*. The *Serial Ports* menu features four tabs for up to four virtual serial connections per VM. Click through each tab, *Port 1* through *Port 4*, and verify the checkbox labeled *Enable Serial Port* is unchecked.

USB brings up the USB controller menu. Remember that by default, VirtualBox only provides USB 1.1 support. For USB 2.0 or 3.0 support, users will need to install VirtualBox extensions. Fortunately, none of our VMs will require any USB support. Uncheck the *Enable USB Controller* checkbox.

The *Shared Folders* option is next. Folder sharing allows the host and virtual machine to share files. The lab environment does not make use of shared folders, and students are recommended to not enable any. Observe the window labeled *Shared Folders*, and confirm there are no shared folders enabled.

Finally, we come back to *Network*. This menu allows students to enable up to four network interfaces on a virtual machine, define what networks they are attached to, and various, other advanced networking features. Students will need to perform a number of complex tasks in this menu, described in the section below.

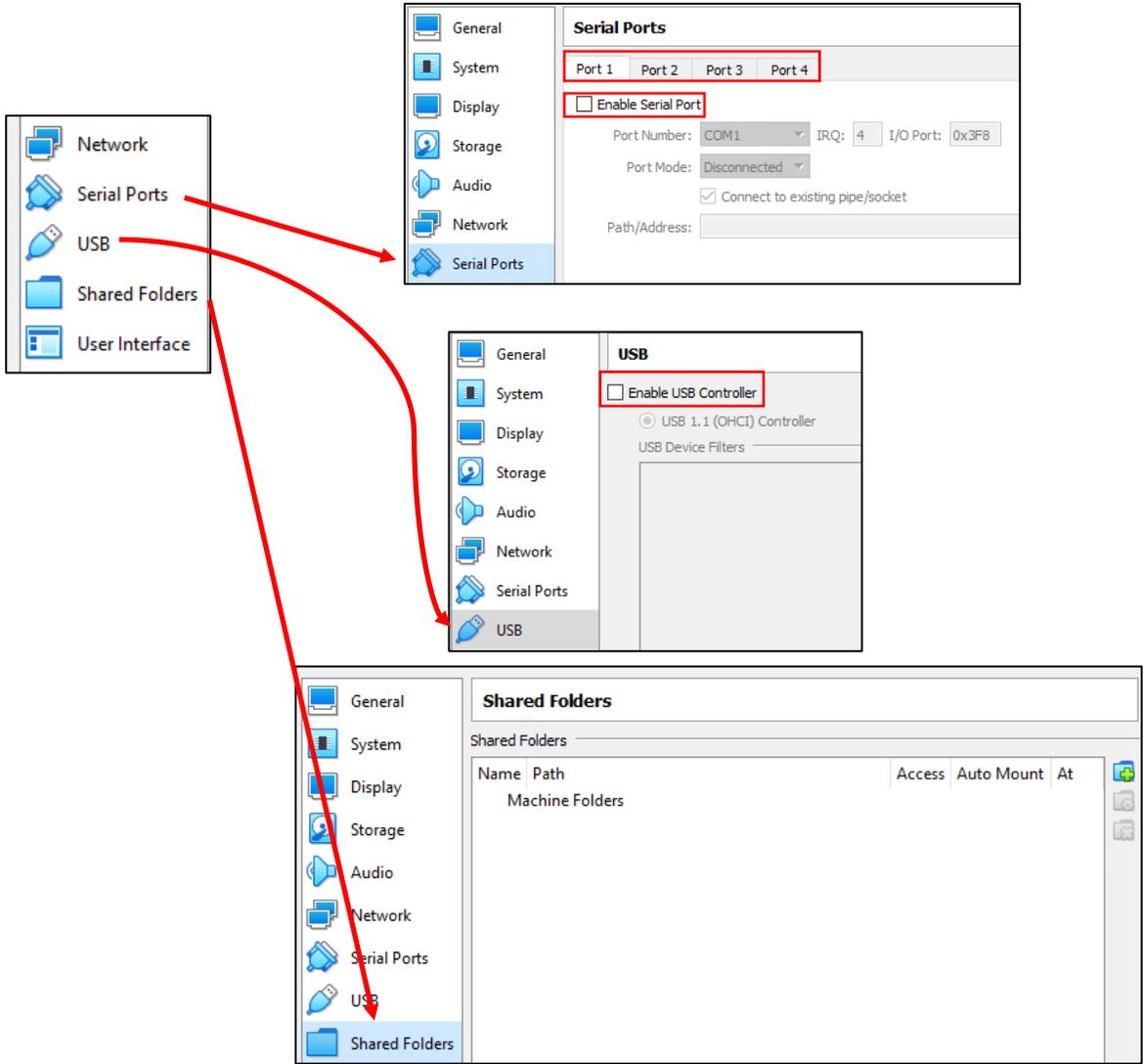
Note: I felt the need to put this in a note box, because this still gets ME once in a while! Students will need to click the *OK* button (usually located in the bottom right corner of every menu) in order for any configuration settings to apply to their virtual machines. After clicking *OK*, the VM settings menu will close. It is not required to click the *OK* button after modifying settings in every individual menu. For example, a user could uncheck the *Enable Audio* option under *Audio*, then navigate to *USB* and uncheck *Enable USB Controller*, then click *OK* on the *USB* menu screen to apply both settings at once.



Continued to 10-18

10-17: Access the [VM name] – Settings menu. Under General, ensure both the Shared Clipboard, and Drag'n'Drop settings are Disabled. Under System, Note the Boot Order for later, then change the Pointing Device to PS/2 Mouse. Under Storage, Click the blue CD icon under Storage Devices, then click the small blue optical disk icon under Attributes, followed by the Choose a disk file option. Browse to the location of the decompressed pfSense ISO, and select it. In Audio, uncheck the Enable Audio checkbox.

Continued from 10-17

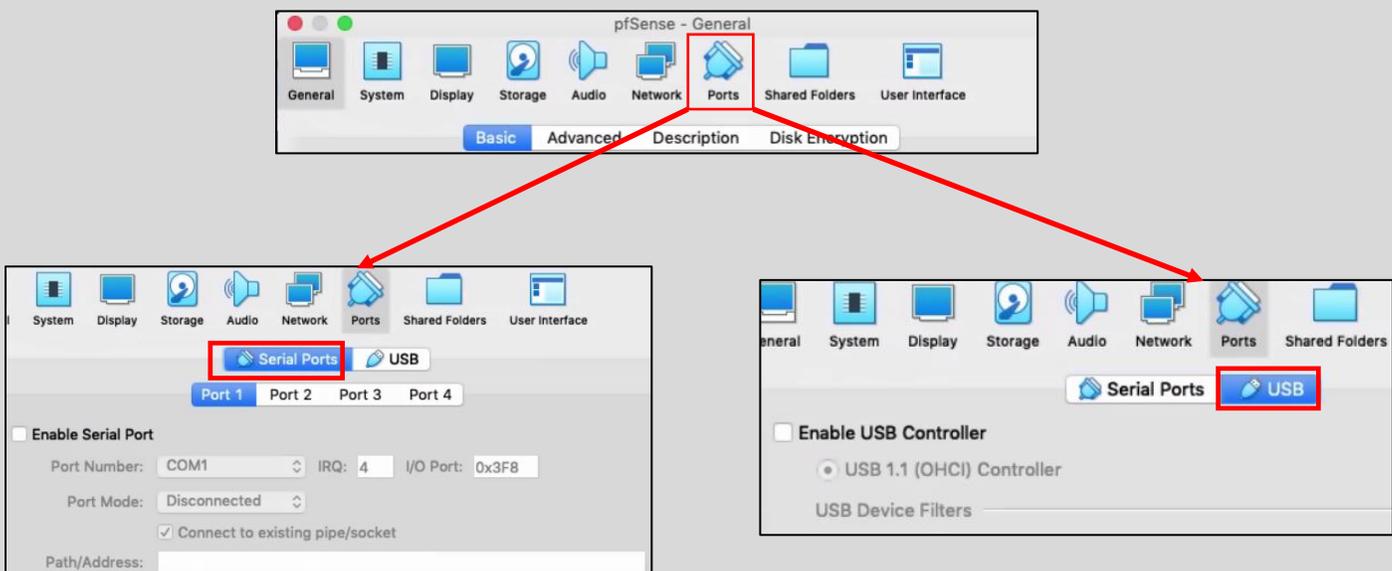


10-18: Navigate to *Serial Ports* and under the tabs labeled *Port 1* through *Port 4*, confirm the *Enable Serial Port* checkboxes are all unchecked. Under *USB*, uncheck the *Enable USB Controller* checkbox. For the *Shared Folders* setting, confirm there are no shared folders configured. The *Network* option will be covered separately in the next section.

MacOS – Different for the Sake of Being Different

MacOS users have a VM settings window that is just slightly different from the version Windows and Linux users get. It's similar enough to where the MacOS users should be able to comprehend the instructions provided and apply them, but just different enough to where I believe it's a good idea for me to call out where the differences are, to make absolutely sure MacOS users can follow along:

- The title of the VM settings window changes based on the menu option highlighted. When the settings window is first opened, users are in the *General* settings menu. The title of the window will be *[VM name] – General*.
- The configuration options menu depicted on the left side of the window for Windows and MacOS users is instead stretched across the top of the VM settings window (See fig. 10-18). Most of the menu options are the same except...
- Instead of there being two distinct menu options for Serial Ports and/or USB ports, there is a *Ports* menu option. When highlighted, there are two tabs labeled *Serial Ports* and *USB*. Left clicking will highlight the tab and change the available menu options accordingly (Also depicted in fig. 10-19).



10-19: This illustration depicts the Virtualbox VM settings menu on MacOS. The menu options are listed horizontally. Additionally, the *Serial Ports* and *USB* menu options are gathered into the *Ports* menu option, that has two tabs – *Serial Ports* and *USB* that can be clicked to access those options, respectively. With this information, MacOS users should be able to utilize the instructions above in section 10.6.2 to configure their VM settings.

10.6.2.1 Virtual Machine Network Settings

This section is dedicated to teaching students how to navigate the *Network* option for the VirtualBox VM settings menu for the pfSense virtual machine. The network configuration for the pfSense VM is extremely important, because it is meant to be situated between multiple virtual networks in our lab. pfSense has the job of enforcing network boundaries and segmentation. Getting readers familiar with the various network configuration options ensures that the pfSense VM is set up correctly, and provides them the knowledge necessary for setting up the other lab VMs later in this chapter.

Navigate to the *Network* option under the pfSense VM's *Settings* menu. Notice there are four tabs in the menu labeled *Adapter 1* through *Adapter 4*. The first setting in each tab is a checkbox labeled *Enable Network Adapter*. Ensure this checkbox is checked for Adapters 1, 2 and 3, and unchecked for *Adapter 4*. **Adapter 4 should never be enabled.**

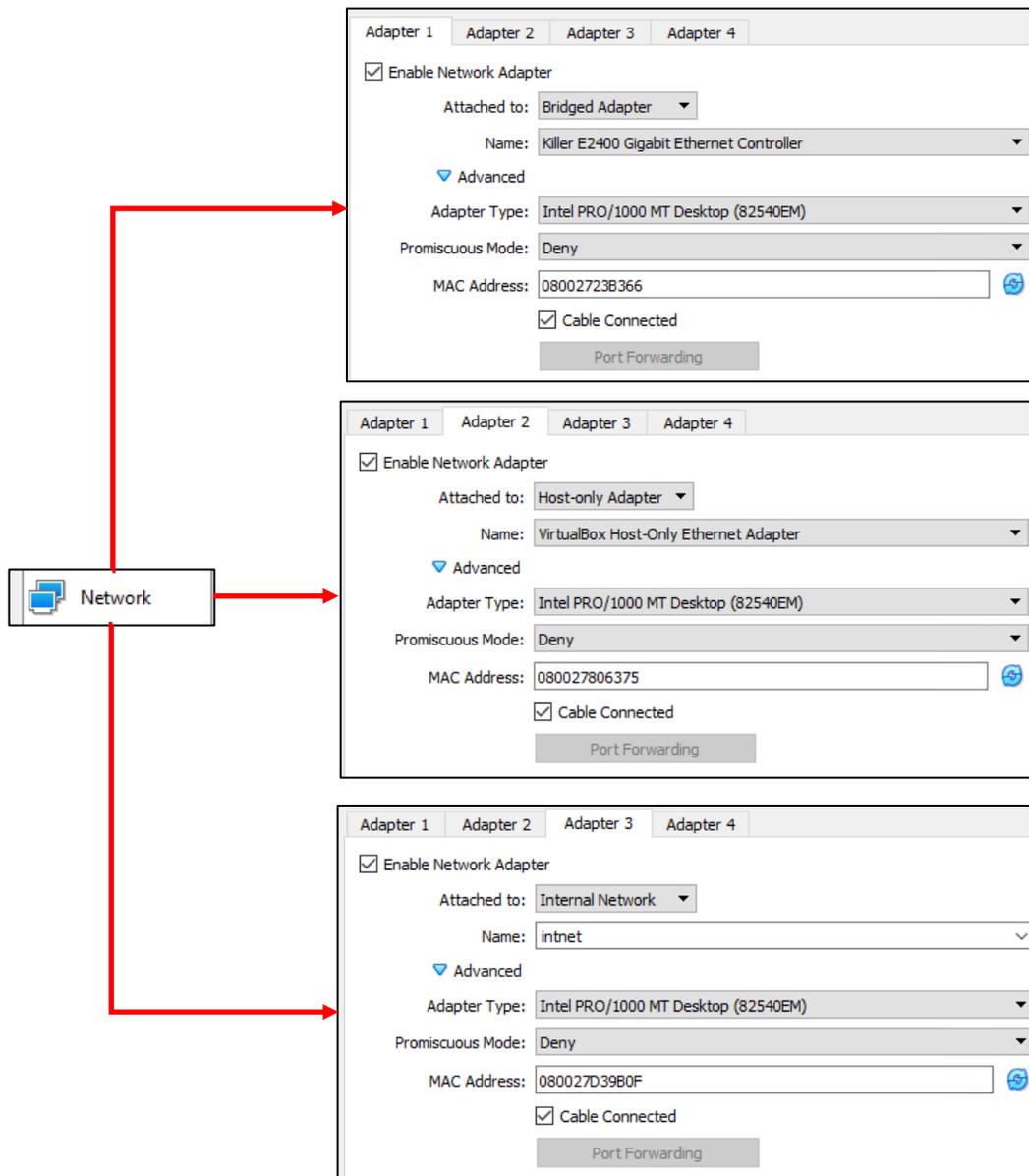
The setting immediately under the checkbox is a drop-down menu labeled *Attached to*. Clicking the menu reveals a variety of network segments user may connect virtual adapters to. *Adapter 1* is going to be the pfSense WAN interface connected to our bridged network – your local physical network. Select the *Bridged Adapter* setting. Underneath, the drop-down labeled *Name* becomes selectable. If students have more than one network interface installed on their host system, they may choose which network interface *Adapter 1* will bridge to. Make use of network interface monitoring tools and network troubleshooting commands available on your host operating system to verify which interface is the correct choice to bridge to.

Click on the option labeled *Advanced* to display a variety of advanced network configuration options and features. There is no need to change any of these settings. **Take note of the input box labeled MAC Address. Record the contents of the input box. Also, note the location of the Promiscuous Mode drop-down. This setting will be important later.**

Adapter 2 will be the pfSense LAN interface connected to the management network. In the *Attached to* drop-down, select *Host-only Adapter*. Windows users should confirm *VirtualBox Host-Only Ethernet Adapter* is selected in the *Name* drop-down box, while Linux and MacOS should confirm that *vboxnet0* is selected. Click *Advanced*, and just like with *Adapter 1*, **record the contents of the MAC Address input box.**

Adapter 3 will be the pfSense OPT1 interface connected to the IPS 1 network. *Attached to* should be set to the *Internal Network* setting. The *Name* field turns into an input box. The default name provided is *intnet*. Click *Advanced*, and **record the MAC Address input box.**

If students are lost or confused, refer to figure 10-20, as well as the sidebar conversation, *Noting the Notable*. Be aware that the contents of the *MAC Address* input box will differ, as will the *Name* field for *Adapter 1* and/or *Adapter 2* (based on your hardware and operating system). Once finished, click *OK* to apply the configuration changes, and exit the VM settings menu.



10-20: Under the *Network* setting for the pfSense VM, ensure the *Enable Network Adapter* checkbox is checked for Adapters 1, 2 and 3. For *Adapter 1*, select *Bridged Adapter* in the *Attached to* drop-down menu. Click the *Advanced* option, and in the configuration settings that appear, Record the contents of the *MAC address* field, and ensure that the *Cable Connected* checkbox is checked. Repeat this process for *Adapter 2*, setting the *Attached to* drop-down to *Host-only Adapter*, and *Adapter 3*, setting it to *Internal Network*. Confirm that the *Name* input box below *Internal Network* is set to `intnet`.

Please be aware that the *Name* field on *Adapter 1* and/or *Adapter 2* may differ based on the host OS and/or physical network adapter being bridged to. Students should document all three MAC addresses. Note that they belong to the pfSense VM, and document the networks in which they are attached.

Noting the Notable

I can't overstate the value of documenting your lab network properly. Use whatever note-taking methods you prefer – paper and pen, Evernote, text editors, personal wikis, databases, spreadsheets, etc. Document the name of the VM, Operating system, the number of CPU cores allocated, RAM, Disk size, number of network adapters, network segments they are attached to, and their MAC addresses. This is called *asset management*, and it's an important habit to cultivate. Here is a template you can use for documenting your VMs:

VM Name:
Operating System:
CPU Cores:
RAM:
Disk Size:
Virtual Network Adapters:
Network Adapter #:
-Network Segment:
-MAC Address:
<Repeat for each network adapter>
Additional Notes:

And as an example, here is my pfSense VM entry:

VM Name: pfSense
Operating System: pfSense (FreeBSD)
CPU Cores: 1
RAM: 512MB
Disk Size: 5GB
Virtual Network Adapters: 3
Network Adapter 1:
-Network Segment: Bridged/WAN
-MAC Address: 08:00:27:23:B3:66
Network Adapter 2:
-Network Segment: Management/LAN
-MAC Address: 08:00:24:80:63:75
Network Adapter 3:
-Network Segment: IPS 1/OPT1
MAC Address: 08:00:24:D3:9B:0F
Additional Notes: Lab firewall. Provides NTP, DNS, DHCP,
and HTTP proxy services.

Do this for every single virtual machine you add to your lab environment. Keep track of systems added or removed from the lab network. Always be aware of what's running on your networks. If you can do these things, you'll be better at asset management than most of the Fortune 500.

10.6.3 First Boot and OS Installation

With the pre-boot configurations completed, students are now ready to install the pfSense firewall distribution to their virtual machine. In the main *VirtualBox Manager* window, highlight pfSense, and click the big green *Start* arrow.

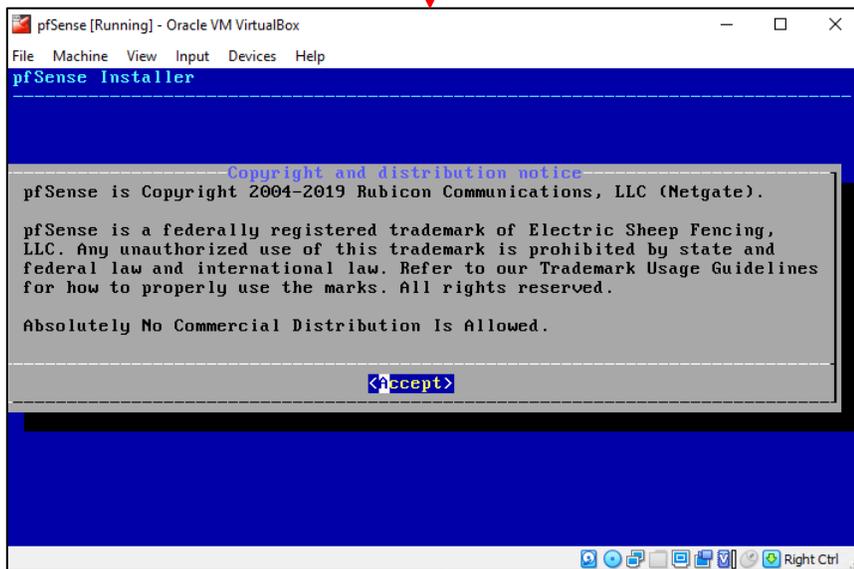
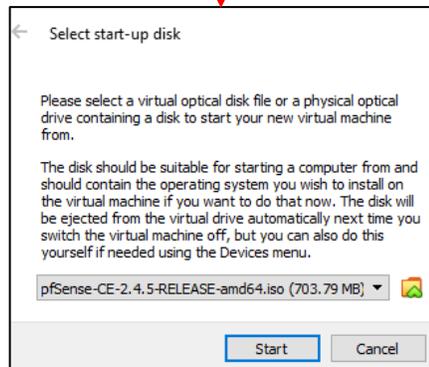
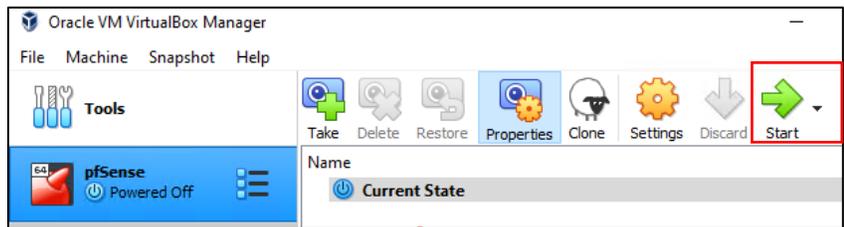
Two new windows will pop up. The window in the foreground is labeled *Select start-up disk*. This is a feature to make sure a bootable ISO image has been selected, if the hypervisor detected that the virtual hard disk does not contain an operating system. We already mounted the pfSense ISO on the virtual CD/DVD drive through the pfSense VM's *Storage* settings, so click the *Start* button to continue.

The background window will take over. This is the virtual console for the pfSense VM. Think of this window as a direct keyboard, video, and mouse connection to the virtual machine while it is running. You'll notice a lot of text flying by as the VM boots from the installation ISO. Eventually students will reach the pfSense Installer. The first screen shows the *Copyright and distribution notice* for the software. Click anywhere in the virtual console window, and hit Enter to accept the software terms and conditions.

Next is the *Welcome* screen for the OS Installer. The option *Install pfSense* should be highlighted by default, but if not, use the arrow keys on your keyboard to select it, then hit enter. The next screen, titled *Keymap Selection* appears. If students are from a region of the world with a unique keyboard layout, they will need to search for and select it. Otherwise, select *Continue with default keymap* to use the US keymap, and hit enter. Next is the *Partitioning* screen. Partitioning is used to tell the installer how much and what portion of the disk to allocate. Since this is a virtual machine, and the disk is relatively tiny (5GB), select *Auto (UFS) Guided Desk Setup* and press enter to tell the installer to use all of the available disk space.

The installer handles formatting the disk and copying the operating system files over. The next screen, titled *Manual Configuration* asks if you want a command shell to manually edit any operating system files before closing the installer. Select *No*, and hit enter again. Finally, on the *Complete* screen, select *Reboot*, and hit enter. Congrats! You just installed the pfSense firewall distribution to your pfSense virtual machine.

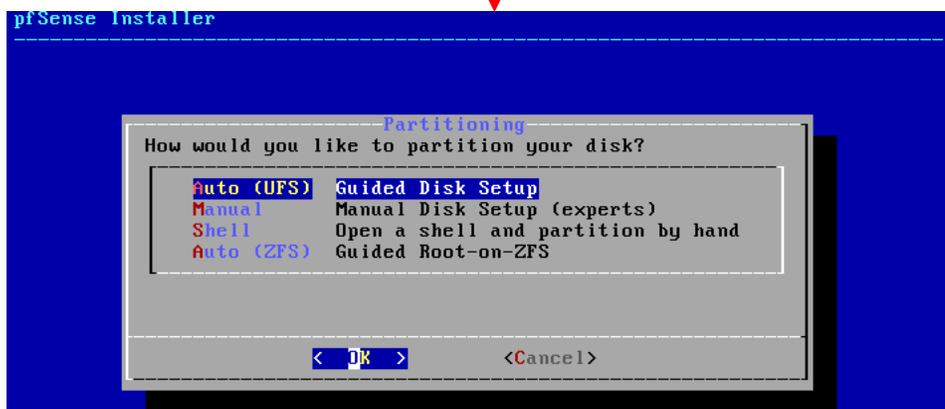
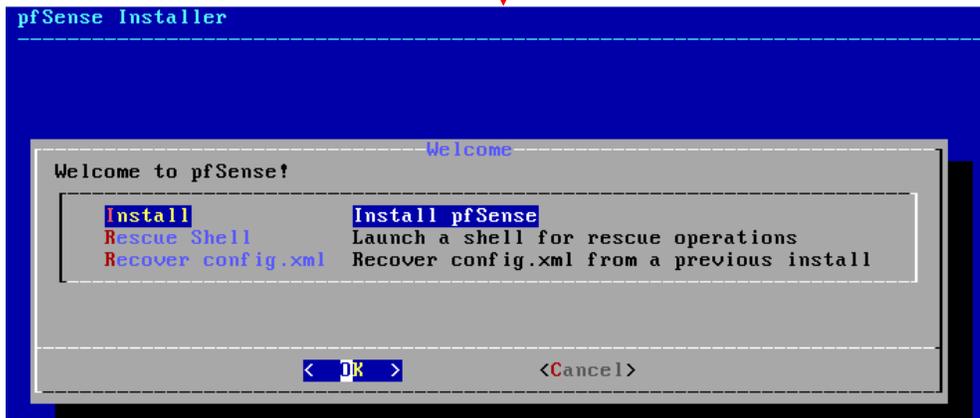
While the system is rebooting, close the virtual console window. A small window will appear asking what you would like it to do. Select the option *Power off the machine*, and click *OK*. Take note of this: Starting a virtual machine opens the virtual console. Closing the console will trigger options to shut down the virtual machine. There are other ways to start and stop VMs, but this is the easiest.



Continued to 10-22

10-21: Highlight the pfSense VM in the VirtualBox Manager window, then click the big green *Start* button to boot the virtual machine. A pop-up labeled *Select start-up disk* will appear, asking students to select an ISO to boot from. Since this task was already performed, click the *Start* button to proceed, and the virtual console should appear. After a moment the copyright and distribution notice for pfSense should appear. Press enter to accept and proceed with the rest of the installer.

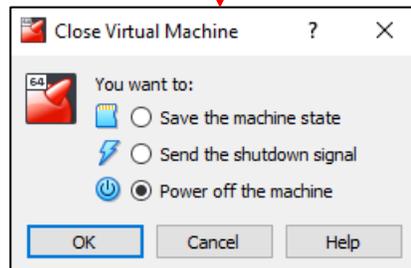
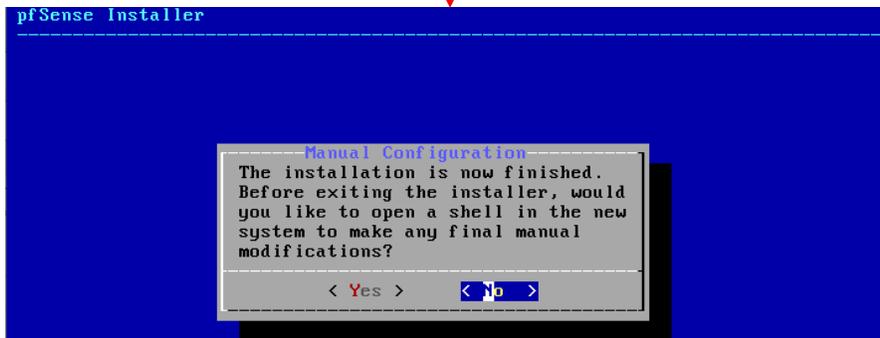
Continued from 10-21



Continued to 10-23

10-22: On the *Welcome to pfSense* screen, select *Install PfSense*, and hit enter. On the *Keymap Selection* screen, choose the appropriate keymap for your region with the arrow keys, then hit enter to proceed. Next, on the *Partitioning* screen, select *Auto (UFS)* then hit enter to install pfSense to the virtual machine's disk.

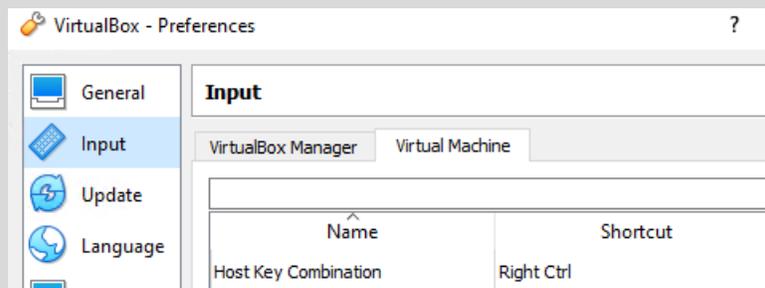
Continued from 10-22



10-23: After a moment or two, the installer finishes copying files to the pfSense VM's virtual disk. On the *Manual Configuration* screen, highlight *No* and hit enter to continue. Then on the *Complete* screen, highlight *Reboot* and hit enter to finish the installation process. While the pfSense VM is rebooting, close the virtual console window, and a window labeled *Close Virtual Machine* will appear. Select the *Power off the virtual machine* radio button, then click the *OK* button to shut down the pfSense VM.

Virtual Machines Ate My Neighbors Input

When a user clicks on the virtual machine console, that window grabs all of the input from the mouse and keyboard. That means that if there are other applications running on your host system you want to interact with, you have to tell the virtual console to "let go" of the mouse and keyboard first. VirtualBox uses a special key binding called the *Host Key Combination* to signal to the virtual console that you have other things to do. On Windows and Linux, this is the Ctrl key on the right side of your keyboard. On MacOS, because they have to be different™, this key is the left meta key (the thing that looks like a clover). You can change the key that controls this behavior in the VirtualBox Manager's *Preferences* menu, under the *Input* option, on the *Virtual Machine* tab.



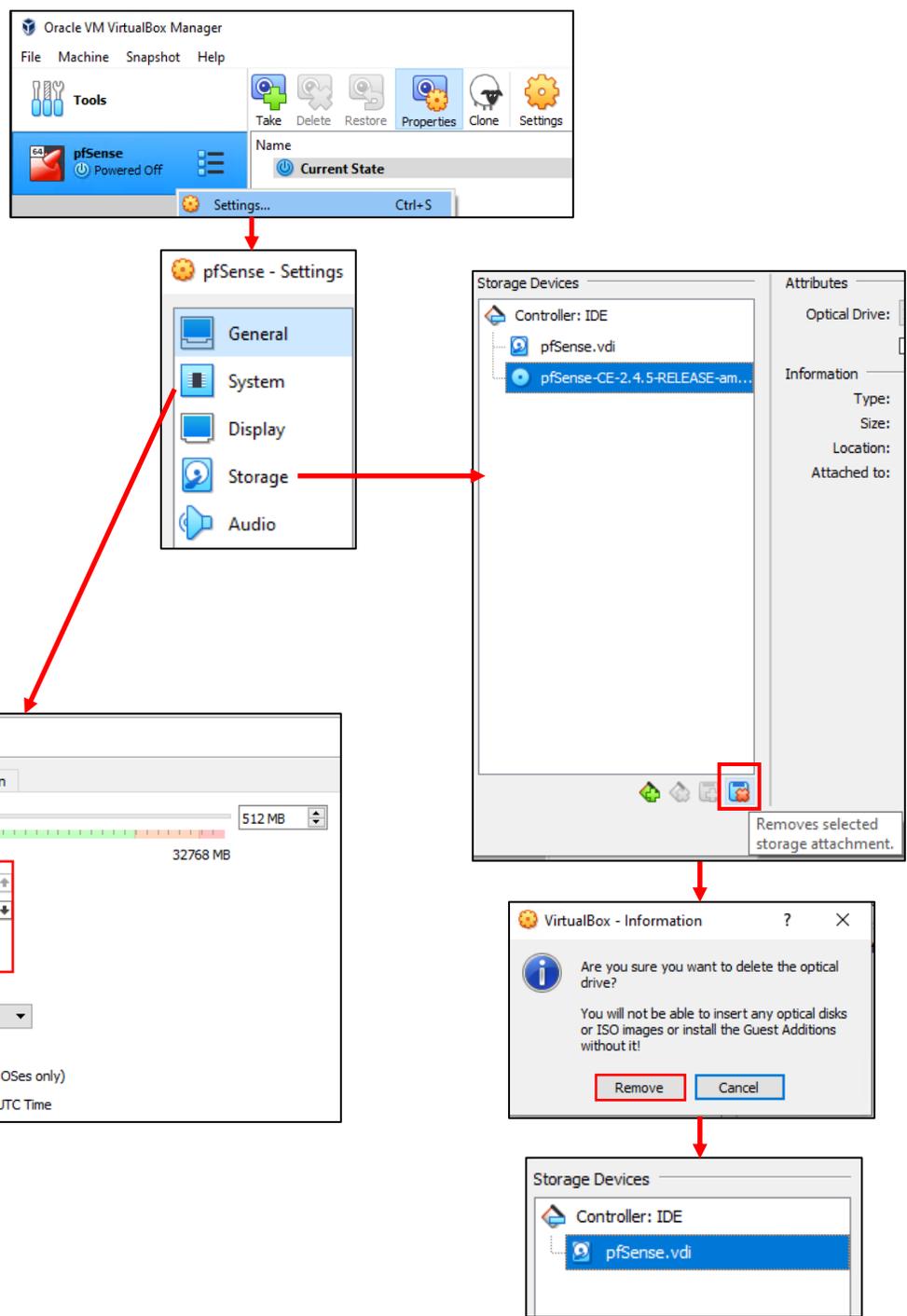
10-24: Look for the *Host Key Combination* option under *Preferences > Input > Virtual Machine* tab if you need to know what key releases control from the virtual console, or to rebind the shortcut combo.

10.6.4 Virtual Machine Settings (Part 2)

In this section, students will be making a couple of minor changes to the *System* and *Storage* options in the virtual machine settings menu for pfSense. With the pfSense VM in the *Powered Off* state, right click on it in the *VirtualBox Manager* window, and select *Settings* to open up the virtual machine settings menu again.

Once in the menu, navigate to the *System* option. Recall in section 10.6.2 the *Boot Order* window in the *Motherboard* tab. The boot order defines what devices the virtual machine will attempt to check for a bootable operating system, and in what order it will check them. We want to modify the boot order to where our virtual machine will only be allowed to boot from the hard disk, and ensure the hard disk is the first device that is checked when the VM is powered on. Uncheck the checkboxes next to both *Floppy* and *Optical*, then left click on *Hard Disk* to highlight it, then click the small upwards facing arrow right of the *Boot Order* window until it is the first item listed in the window. If performed correctly, the *Hard Disk* item will be the only item in the list with a checkbox, and it will be the first item at the top of the list. Once finished, click on the *Storage* option to bring up the VM storage settings.

Under the *Storage Devices* column, click on the blue disc icon that looks like a CD to highlight it. This is the virtual CD/DVD drive we used to install pfSense to our VM. At the bottom of the *Storage Devices* window are four small icons – two look like diamonds, with a + and an x over them, while the other two look like squares with the same + or x over them. Hover over the square with the x over it and a box will pop up that reads, *Removes selected storage attachment*. Click on square to remove the virtual CD/DVD drive. A pop-up box appears asking if students are sure you want to remove the optical drive. Click the *Remove* button to continue. The *Storage Devices* listing updates to reflect that we no longer have a virtual CD/DVD drive. Click the *OK* button in the bottom right portion of the VM settings window to apply both the *System* and *Storage* settings, and exit the menu.



10-25: With the pfSense VM powered off, Open its settings menu. Under the *System* options on the *Motherboard* tab, modify the *Boot Order* list to where the *Hard Disk* is the first item at the top of the list, and the only item with a checkbox. Next, navigate to *Storage* settings, and under the *Storage Devices* column, remove the virtual CD/DVD drive. When finished click *OK* to apply these changes and exit the settings menu for the pfSense VM.

10.6.5 pfSense Command-Line and initial interface configuration

In this section, readers will navigate the command-line interface of their pfSense virtual machine to perform essential setup tasks. Once finished, students will jump to chapter 14 to finish configuring their pfSense VM using its web interface, the webConfigurator. Begin by highlighting pfSense in the *VirtualBox Manager*, and clicking the big green start button to power on the VM and bring up its virtual console. After a few moments, the boot process completes and students are presented with the pfSense command-line menu. This menu features a series of configuration and troubleshooting options. Begin by selecting option 1 and hitting enter.

10.6.5.1 The Assign Interfaces Wizard

Users are greeted by the *Assign Interfaces* wizard. This wizard is used to map our virtual machine's network interfaces (*Adapter 1*, *Adapter 2*, and *Adapter 3*) to their pfSense aliases – *WAN*, *LAN*, or *OPT1*. Unfortunately, the operating system itself also has unique names for each of these interfaces, adding another layer of complexity and confusion when trying to perform this task.

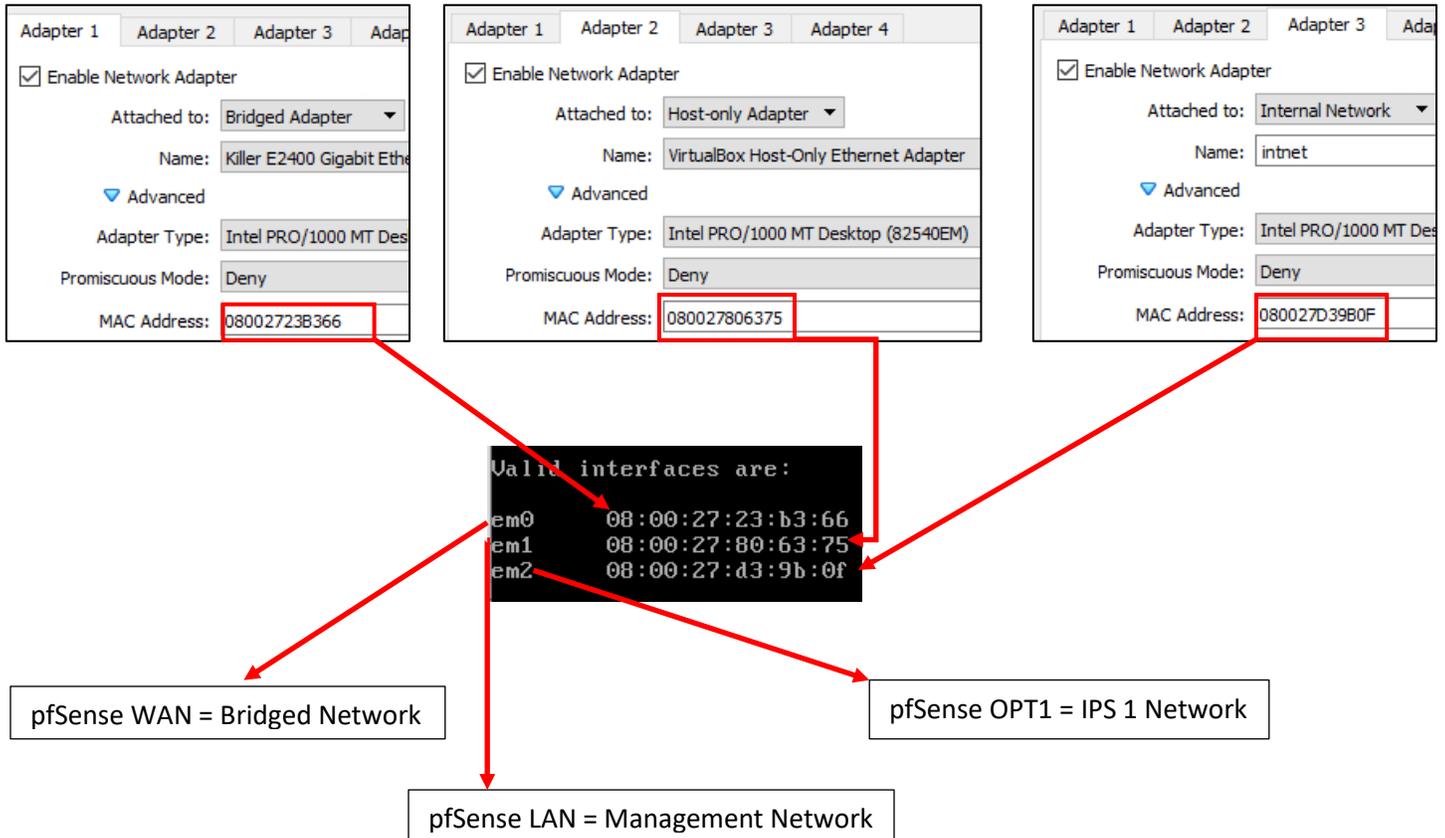
pfSense itself is based on the FreeBSD operating system, and BSD has its own methods for assigning physical (or virtual, in our case) network interfaces an interface name. For example, BSD assigned the network adapters of my virtual machine the interface names *em0*, *em1*, and *em2*. Every network adapter – integrated or not, virtual or physical, wired or wireless – all have a MAC address to uniquely identify them on a local network. We're going to take advantage of that to know for certain which of the three interfaces *em0* through *em2* map to VirtualBox *Adapters 1* through 3, and how they should be assigned as the *WAN*, *LAN* and *OPT1* aliases.

A quick way for readers to determine the interface names for their pfSense installation is through the wizard itself. Upon selecting option 1, a section of text labeled *Valid interfaces are* appears, followed by a series of lines. **Students should have 3 of these lines in total.** These lines provide the interface names, MAC addresses, current operational status, and type of hardware BSD identifies the network interface as (The drivers BSD loaded) for each network interface pfSense was able to detect. Here is an example:

```
Valid interfaces are:
em0 08:00:27:23:b3:66
em1 08:00:27:80:63:75
em2 08:00:27:d3:9b:0f
```

10-26: A portion of the *Assign Interfaces* wizard. Pay attention to the interface names (1) and the MAC addresses for those interface names (2). This information is needed to determine which virtual network segment they are connected to. This in turn allows students to assign the *WAN*, *LAN* and *OPT1* interfaces correctly.

Compare the MAC addresses displayed, to the MAC addresses recorded earlier, and use that information to complete the rest of the Assign Interfaces wizard. A diagram (fig. 10-27) is provided below to help students understand how to correctly perform this mapping process.



10-27: Here we have the network configuration for my pfSense VM, and the output from the valid interfaces table from the *Assign Interfaces* wizard. *Adapter 1* has the MAC Address 08002723B366. Looking at the valid interfaces table, *em0* has the same MAC address, just with colons (:) every 2 characters (the correct notation for MAC addresses). This means that *em0* maps to *adapter 1*, connected to the bridged network. *em0* should be assigned as the *WAN* interface. *Adapter 2*'s MAC address matches the MAC address for *em1*. This means *em1* maps to *adapter 2*, connected to the host-only network – our management network. This means *em1* should be assigned the *LAN* interface. Finally, *adapter 3* matches the MAC address for *em2*. This means *em2* maps to the *intnet* – IPS 1. This means that *em2* should be assigned the *OPT1* interface.

The remainder of this section will aim to guide students through the various questions the wizard will ask (in *italicized* font), and the answers I provided (in *bold* font) based on my lab network and adapter to MAC address mappings. **Students should be aware that this is by and far the most important configuration task for pfSense.** Making sure that the VirtualBox adapters map to the correct pfSense aliases and network segments is absolutely vital to the lab environment working correctly.

Should VLANs be set up now [y|n]? n

*Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): **em0***

*Enter the LAN interface name or 'a' for autodetection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): **em1***

*Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): **em2***

The interfaces will be assigned as follows:

WAN -> em0

LAN -> em1

OPT1 -> em2

*Do you want to proceed [y|n]? **y***

After answering these questions, pfSense will loop back to the main menu.

```

pfSense 2.4.5-RELEASE amd64 Tue Mar 24 15:25:50 EDT 2020
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 5a25c089ce30ec1b5b9e

*** Welcome to pfSense 2.4.5-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0          -> v4/DHCP4: 10.0.0.39/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:
/64
LAN (lan)      -> em1          -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 1

Valid interfaces are:

em0      08:00:27:23:b3:66   (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em1      08:00:27:80:63:75   (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em2      08:00:27:d3:9b:0f (down) Intel(R) PRO/1000 Legacy Network Connection 1.

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.
Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

The interfaces will be assigned as follows:

WAN -> em0
LAN -> em1
OPT1 -> em2

Do you want to proceed [y|n]? y

```

10-28: A selection of screen captures from the Assign Interfaces wizard, stitched together to show the questions the wizard asks, and the responses based on network adapter mappings in *fig. 10-27*.

10.6.5.2 Setting IP Addresses for WAN, LAN, and OPT1

The next task we will need to perform on the pfSense command-line is assigning IP addresses to the *WAN*, *LAN*, and *OPT1* interfaces using the *Set interface(s) IP address* wizard. Most students will have their host system connected to a home or enterprise network where DHCP is available, and just about anything that requests an IP address lease will get one with no problems. That *should* include the pfSense *WAN* interface bridged to that network. This means the *WAN* interface should already have an IP address, subnet mask, default gateway (and usually, DNS servers to forward DNS requests to) automatically provided (if this is not the case, see the sidebar discussion, *Help! The WAN Interface has no IP Address*, for some troubleshooting pointers). That means we should only have to run through the *Set interface(s) IP address* wizard twice – once for the *LAN* interface, and once for the *OPT1* interface. Select option 2 from the pfSense menu to get started.

Similar to the previous section (10.6.5.1), the remainder of this section is going to consist of the questions the *Set interface(s) IP address* wizard will ask students (*italicized*), and the correct answers for the *LAN* and *OPT1* interfaces (in **bold**), followed by an illustration depicting the same questions and answers.

LAN interface:

Available interfaces:

- 1 – WAN (*[interface name] – [dhcp/dhcp6/static address configuration]*)
- 2 – LAN (*[interface name] – static*)
- 3 – OPT1 (*[interface name]*)

Enter the number of the interface you wish to configure: **2**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.1.1**

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense

e.g. 255.255.255.0 = 24

255.255.0.0 = 16

255.0.0.0 = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.1.10**
Enter the end address of the IPv4 client address range: **172.16.1.254**
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24

You can now access the webConfigurator by opening the following URL in your web browser:

<https://172.16.1.1>

Press <ENTER> to continue. <ENTER>

```
Enter the number of the interface you wish to configure: 2
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 172.16.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.1.10
Enter the end address of the IPv4 client address range: 172.16.1.254
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
      https://172.16.1.1/

Press <ENTER> to continue.
```

10-29: Screen captures from the *Set interface(s) IP address wizard*, stitched together to show the questions the wizard asks, and the responses for the LAN interface based on network adapter mappings in *fig. 10-27*.

Here is an abridged set of *questions* and **answers** for the *OPT1* interface:

OPT1 interface (abridged):

Available interfaces:

- 1 – WAN (*[interface name] – [dhcp/dhcp6/static address configuration]*)
- 2 – LAN (*[interface name] – static*)
- 3 – OPT1 (*[interface name]*)

Enter the number of the interface you wish to configure: **3**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.2.1**

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**

Enter the start address of the IPv4 client address range: **172.16.2.10**

Enter the end address of the IPv4 client address range: **172.16.2.254**

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

```

Enter the number of the interface you wish to configure: 3

Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 172.16.2.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new OPT1 IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new OPT1 IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on OPT1? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.2.10
Enter the end address of the IPv4 client address range: 172.16.2.254

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to OPT1...
  Reloading filter...
  Reloading routing configuration...
  DHCPD...

The IPv4 OPT1 address has been set to 172.16.2.1/24
Press <ENTER> to continue.

```

10-30: Screen captures from the *Set interface(s) IP address* wizard, stitched together to show the questions the wizard asks, and the responses for the *OPT1* interface based on network adapter mappings in *fig. 10-27*.

After running the wizard again for the *OPT1* interface, students should have an IP address for the *WAN*, *LAN* and *OPT1* interfaces. Additionally, DHCP ranges should be assigned for the *LAN* and *OPT1* interfaces. We're just about ready to move to the webConfigurator, but before doing so, lets run some network connectivity tests first.

```

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.0.197/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:fe23:b366
/64
LAN (lan)     -> em1      -> v4: 172.16.1.1/24
OPT1 (opt1)   -> em2      -> v4: 172.16.2.1/24

```

10-31: The interface information portion of the pfSense command-line menu should look something like this, if all the interfaces are configured correctly.

What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?

Unfortunately, I have no way of knowing what network ranges students use at home, so it's entirely possible your physical network may already be using one of the ranges I'm asking you to configure for your lab environment (e.g., 172.16.1.0/24, or 172.16.2.0/24). **To avoid network conflicts on your home network, maybe try these alternate configurations for the *Set interface(s) IP address wizard*:**

Alternate LAN configuration:

LAN interface IP address: 172.16.11.1
Subnet mask bit count: 24
DHCP start address: 172.16.11.10
DHCP end address: 172.16.11.254

Alternate OPT1 configuration:

OPT1 interface IP address: 172.16.12.1
Subnet mask bit count: 24
DHCP start address: 172.16.12.10
DHCP end address: 172.16.12.254

If your lab network is connected to a school or enterprise network using the entire 172.16.0.0/12 allocation, things may be a little more complicated. It may be best to use one of the other RFC1918 network allocations instead, such as 192.168.0.0/16, or 10.0.0.0/8. Why? Enterprise networking can become complicated, either due to growth over time, legacy configurations, or work-arounds to problems accrued over time. You don't want to troubleshoot network problems on your host system, nor do you want the IT ops team coming to your desk over a network outage that could've been avoided. **Here are some alternate configurations for the *Set interface(s) IP address wizard* if you need to avoid using 172.16.0.0/12 entirely:**

Alternate LAN configuration 1:

LAN interface IP address: 10.0.11.1
Subnet mask bit count: 24
DHCP start address: 10.0.11.10
DHCP end address: 10.0.11.254

Alternate OPT1 configuration 1:

LAN interface IP address: 10.0.12.1
Subnet mask bit count: 24
DHCP start address: 10.0.12.10
DHCP end address: 10.0.12.254

Alternate LAN configuration 2:

LAN interface IP address: 192.168.11.1
Subnet mask bit count: 24
DHCP start address: 192.168.11.10
DHCP end address: 192.168.11.254

Alternate OPT1 configuration 2:

LAN interface IP address: 192.168.12.1
Subnet mask bit count: 24
DHCP start address: 192.168.12.10
DHCP end address: 192.168.12.254

Substituting Instructions for Your Chosen Network Ranges

Keep in mind you don't have to use the alternate configurations recommended above. If students have some experience with networking and subnetting, they're welcome to use any network range that suits them. These are just some suggestions to help those who are not quite as experienced, and want to avoid network conflicts.

As a final reminder, **the remaining sections, chapters, and configuration steps will all assume that readers are using 172.16.1.0/24 for the LAN network and 172.16.2.0/24 for the OPT1 network.** This means you will have to mentally substitute steps and commands for the network range you are using instead.

For example, the lab network diagram in chapter 6 has the Kali VM on the IPS 1 (OPT1) network, with an IP address of 172.16.2.2. If you are using an alternate network configuration for the OPT1 network, say 192.168.12.0/24, then the Kali VM's IP address should be 192.168.12.2. If I say "*run the command ssh username@172.16.2.2 to connect to the kali VM*", you'll have to mentally substitute that with `ssh username@192.168.12.2` instead. As another example, firewall rules denying access to or from 172.16.2.3 (Metasploitable2) should be created for 192.168.12.3 instead. Keep this in mind as you continue to build your lab network!

Help! The WAN Interface has no IP Address

If the WAN interface of your pfSense VM has no IP address, consider some of the following to help with troubleshooting:

-No DHCP – It's pretty rare, but perhaps the WAN interface is bridged to a network without DHCP. This just means that you'll have to run the *Set interface(s) IP address* wizard to manually configure the WAN interface IP address, subnet mask, and default gateway. I've already listed the questions the wizard asks, and provided the answers for the LAN and OPT1 interfaces, but since I have absolutely no idea what IP address and subnet mask is assigned to your local physical network, I cannot tell you what you need to enter for the wizard.

If you don't know either, ask a network administrator or whoever is responsible for your network to assist you. Note that if required to manually configure these settings here, practically all of the tasks that require DNS to be configured (e.g., network connectivity tests, and checking for updates on your VMs) will not work until DNS server addresses are configured. This can be done via the webConfigurator, and will be covered shortly.

-Bridged to the wrong host adapter – Another possibility is perhaps the VirtualBox Network adapter may be bridged to the wrong physical network adapter on your host system. Check out [section 10.6.2.1](#) (pp. 224-226) to determine which physical interfaced the VirtualBox Bridged interface is attached to (and change it, if necessary), Then check out [section 10.6.5.1](#) (pp. 234-237) to make absolutely sure that the pfSense WAN interface was mapped to the correct VirtualBox network interface.

-NAC Interference – If you are network security enthusiast at home or connected to an enterprise network, NAC (network access control) may be preventing the WAN interface from obtaining an IP address. Back in chapter 4, [section 4.1.2, NAT Networking \(and Port Forwarding\)](#) (pp. 46-47), readers learned how network address translation works, and how in situations like this, you may be forced to use VirtualBox's NAT network options to work around network security. If you suspect the WAN interface is being blocked, you can try editing the pfSense VM's *Network* settings for the adapter attached to the bridged network, and instead attach it to the NAT network. For example, [fig. 10-27](#) shows *adapter 1* has *Attached To* set to *Bridged Adapter*. Click the drop-down menu select *NAT*, then click *OK*. Reboot the pfSense VM. It should have an IP address from the NAT network, but that doesn't mean it has network connectivity. After you finish configuring the remaining network interfaces (e.g., LAN and OPT1), make sure to check your network connectivity with the network troubleshooting commands provided in the section below

Note: If this doesn't work, or attempting to subvert network access controls would otherwise get students in trouble, consider talking to your network/systems administrator and seeing if you can get DHCP allocation and/or necessary exceptions put into place. **Don't violate acceptable use policies, and don't break the law.**

10.6.6 Testing Internet Connectivity using Shell commands

Select option 8, labeled *Shell* in the pfSense menu. Doing so will open up a command-line (bash) shell. Run these 3 commands, and observe their output:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

```
Enter an option: 8

[2.4.5-RELEASE][root@pfSense.localdomain]/root: ping -c 4 www.google.com
PING www.google.com (172.217.6.100): 56 data bytes
64 bytes from 172.217.6.100: icmp_seq=0 ttl=54 time=24.496 ms
64 bytes from 172.217.6.100: icmp_seq=1 ttl=54 time=22.714 ms
64 bytes from 172.217.6.100: icmp_seq=2 ttl=54 time=21.638 ms
64 bytes from 172.217.6.100: icmp_seq=3 ttl=54 time=19.490 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 19.490/22.084/24.496/1.813 ms
[2.4.5-RELEASE][root@pfSense.localdomain]/root: nslookup www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.6.100
Name:   www.google.com
Address: 2607:f8b0:4009:812::2004

[2.4.5-RELEASE][root@pfSense.localdomain]/root: curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Mon, 01 Jun 2020 02:53:41 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Mon, 01 Jun 2020 02:53:41 GMT
cache-control: private
set-cookie: 1P_JAR=2020-06-01-02; expires=Wed, 01-Jul-2020 02:53:41 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=DdNV16afHrYu25Utm83temwvvrSe6a4UyA3YHz_JKLFzBAv7xrWi8HjSn2-x1PNmxh3EutjAoFBh15hNpxrU72.jpzLLQV0JHJxaOMh5mFyntk5Gae7KUMe2-d1g8I1KloIb7HzOBP_BB4b0sb41t0Tv1zwOdriVE8ndqfygcrN04; expires=Tue, 01-Dec-2020 02:53:41 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

10-32: The output from the `ping -c 4`, `nslookup`, and `curl -I` commands. All three of these commands completed successfully. Pay close attention to the marked sections above. Note that the IP addresses returned for `nslookup` (In the fields labeled *Address*) may vary based on region.

In a nutshell, these three commands are being used to test various forms of internet connectivity for our VM. `ping -c 4 www.google.com` tells pfSense to send 4 (and only 4) ICMP packets to a specific destination, requesting that the destination respond with its own ICMP packets if it has been reached. `nslookup www.google.com` asks our pfSense virtual machine's configured DNS servers to translate a domain name to an IP address for us. Finally, `curl -I https://www.google.com` is being used to test HTTPS connectivity to the internet. The `-I` option tells the command to only return the HTTP Server headers from our request. All we're really interested in is the line of text: `HTTP/2 200`. This is a thumbs up from Google's webserver confirming that they got our HTTP request with no problems.

Students already familiar with DNS basics may have noticed that we are already trying to ping a domain name (`www.google.com`) with our `ping` command. This means, that in order to actually ping the correct destination, our virtual machine will need to make a DNS request to find the IP address of `www.google.com`. That makes the `nslookup` test redundant, right? Well, yes and no. Later in this chapter, as new virtual machines get created, readers will be advised to perform connectivity tests on those VMs as well. However, the pfSense firewall policy is going to be very strict, so ICMP packets outbound from our lab network will be blocked. Due to how DNS works, the `nslookup` check can still be used to make sure VMs can resolve domain names, and the `curl` connectivity test will be more than sufficient to confirm whether or not lab virtual machines have the internet access they require. After performing these commands and confirming internet connectivity, type `exit` to leave the shell.

My connectivity commands failed! Now what?

If students got anything other than output similar to *fig. 10-32* (e.g., request timeouts and/or packet loss for `ping`, timeouts for `nslookup`, no response for `curl -I`), then there are connectivity issues to be sure. Troubleshooting network connectivity is an extremely complex topic. I can't give you a definitive guide for finding the root of your problem, but I can tell you to start with the basics and work your way up – sometimes the cause of your network problems are settings or hardware that was taken for granted.

Checking physical cabling, link lights and physical connectivity to network devices always comes first. As an extension to that, check out the sidebar in section *10.6.5.2 (Help! The WAN interface has no IP address)* for some additional clues. The VM may be bridged to the wrong physical adapter. Some form of network security (e.g., a network firewall) may be preventing your VM from connecting to the internet. Try connecting the bridged VirtualBox adapter to a NAT network instead. The incorrect network adapter may have been chosen to be the WAN interface. Consider re-running the *Assign Interfaces* wizard again, and compare the MAC addresses from the wizard to the MAC addresses of the network adapter in the pfSense VM's Settings menu. See *fig. 10-27* for guidance on confirming that interfaces have been mapped correctly.

If students were required to run the *Set interface(s) IP address* wizard for the WAN interface (No DHCP), or your local network's DHCP server doesn't assign DNS servers automatically, your troubleshooting commands will fail because pfSense has no way of resolving domain names. We will be covering how to manually configure a primary and/or secondary DNS server for pfSense via the webConfigurator shortly.

If your host system is connected to a physical network already using `172.16.1.0/24` or `172.16.2.0/24`, you may be experiencing network conflicts, routing loops, or other weird behavior. Assign different IP addresses and ranges to the LAN and OPT1 networks to avoid network conflicts. See the sidebar discussion in *10.6.5.2* labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?*

Last but not least, check and double check that you entered the commands correctly. Typos matter on the command-line, and BSD will not hold your hand if the command is entered incorrectly. If all else fails, don't be afraid to ask others for guidance.

10.6.7 Finish setting up pfSense

Navigate to [chapter 14, *pfSense Firewall Policy and Network Services*](#), starting on p. 664 and follow the chapter guidance. Once completed, readers will be directed back here to complete their lab environment.

10.7 Create the Remaining Virtual Machines

Welcome back! Now that the pfSense VM is fully functional, it's time to start working on the remaining lab VMs. In this section, users will create three of the four remaining virtual machines via the *Create Virtual Machine* wizard, then adjust the *Settings* of each virtual machine. After the SIEM, IPS and Kali VMs are created and configured, readers will be guided through the operating system installation, and initial setup process for all three VMs. The Metasploitable 2 VM is a unique case, and will be covered separately.

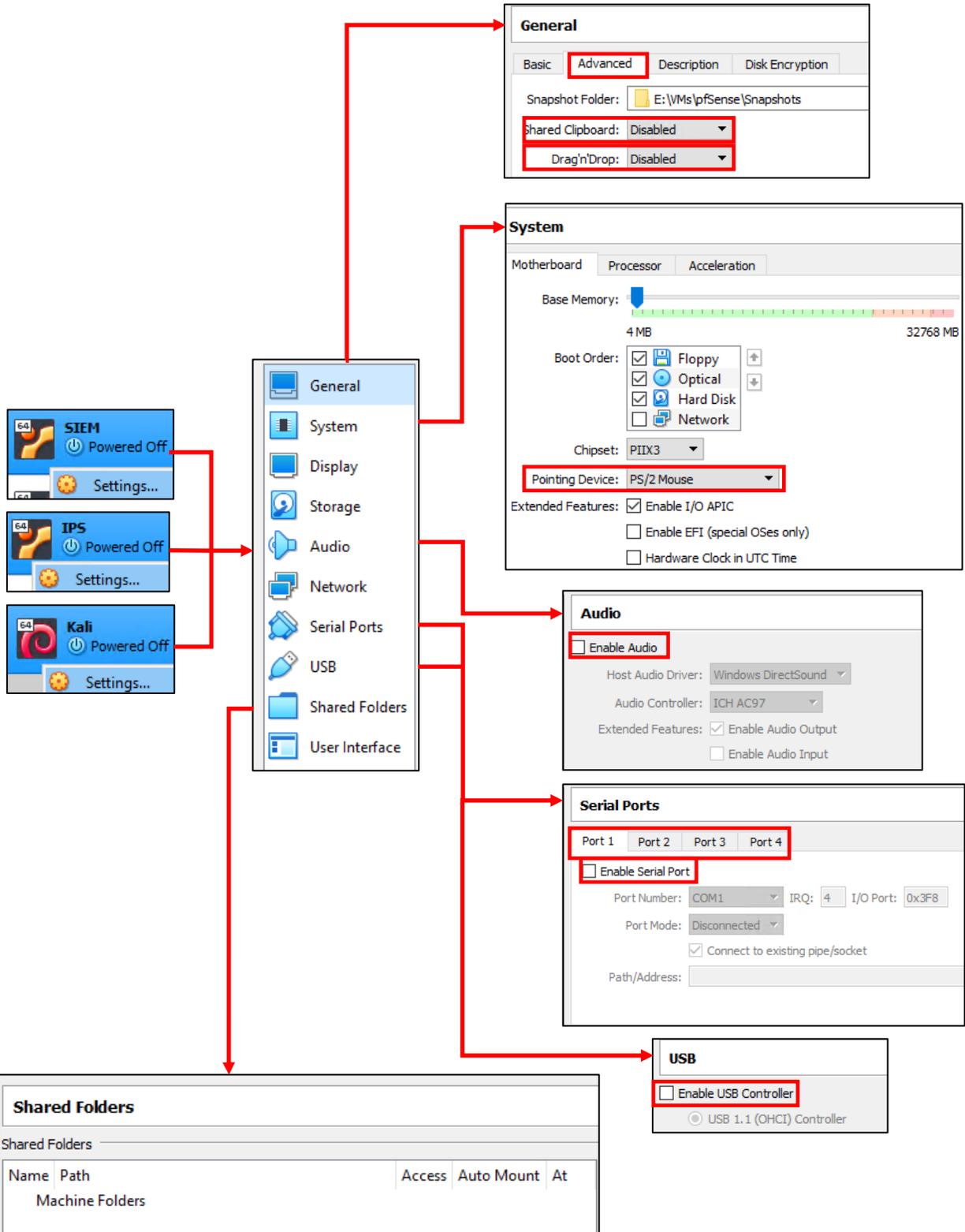
10.7.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali

Run the *Create Virtual Machine* wizard three times, with the settings listed below. Refer back to [section 10.6.1](#) (pp. 216-218) for guidance on how to access and progress through the wizard as needed.

Name	SIEM	IPS	Kali
Type	Linux	Linux	Linux
Version	Ubuntu (64-bit)	Ubuntu (64-bit)	Debian (64-bit)
Memory Size	4GB (4096MB)	4GB (4096MB)	4GB (4096MB)
Hard Disk	80GB VDI, Fixed	80GB VDI, Fixed	80GB VDI, Fixed

Next, students will need to customize the virtual machine settings of all three VMs. Students will be making the same adjustments made for the pfSense VM in [section 10.6.2](#) (pp. 219-223):

General	Ensure both the <i>Shared Clipboard</i> as well as the <i>Drag'n'Drop</i> drop-downs under the <i>Advanced</i> tab are set to <i>Disabled</i>
System	Change the <i>Pointing Device</i> drop-down to <i>PS/2 Mouse</i>
Audio	Uncheck the <i>Enable Audio</i> checkbox
Serial Ports	Verify the <i>Enable Serial</i> checkbox on the <i>Port 1</i> through <i>Port 4</i> tabs are all unchecked
USB	Uncheck the <i>Enable USB Controller</i> checkbox
Shared Folders	Verify there are no shared folders defined



10-33: These are settings that will need to be configured on the SIEM, IPS and Kali VMs. If these settings, and this illustration looks familiar, it's because these are the same settings used for the pfSense VM in section 10.6.2

Students will need to make specific customizations to the *Storage* and *Network* settings of the three virtual machines. As mentioned above, [section 10.6.2](#), and figures 10-16 and 10-17 can be used as a point of reference for performing these configuration changes.

SIEM VM:

Storage: Students will need to mount the Ubuntu Server ISO to the SIEM VM's virtual CD/DVD drive.

Network: Ensure the *Enable Network Adapter* checkbox is checked only on the *Adapter 1* tab.

Adapter 1:

Attached To: *Host-Only Adapter*

Name: *VirtualBox Host-Only Ethernet Adapter* (Windows), *vboxnet0* (Linux/MacOS)

Click *Advanced* button, and record the contents of the *MAC Address* field, and Confirm the *Cable Connected* checkbox is checked. Use this MAC address to create a static DHCP mapping on *LAN* interface of the pfSense VM, and assign it the IP address 172.16.1.3. Refer to chapter 14, [section 14.3.4.1](#) (pp. 690-692) for instructions on how to perform this task, if necessary.

The image shows two screenshots from a virtual machine configuration interface. The top screenshot shows the **Storage** tab with the **Storage Devices** section containing a controller of type **IDE** and a device named **ubuntu-20.04-live-server-amd6...**. The bottom screenshot shows the **Network** tab for **Adapter 1**. The **Enable Network Adapter** checkbox is checked. The **Attached to:** dropdown is set to **Host-only Adapter**, and the **Name:** field is **VirtualBox Host-Only Ethernet Adapter**. In the **Advanced** section, the **Adapter Type:** is **Intel PRO/1000 MT Desktop (82540EM)**, **Promiscuous Mode:** is **Deny**, and the **MAC Address:** is **080027B8BA24**. The **Cable Connected** checkbox is checked. A red box highlights the MAC address. Below these screenshots is a pfSense configuration window titled **Services / DHCP Server / LAN**. It shows a table of **DHCP Static Mappings for this Interface** with the following data:

Static ARP	MAC address	IP address	Hostname	Description
	08:00:27:b8:ba:24	172.16.1.3	SIEM	DHCP mapping for SIEM VM

10-34: Mount the Ubuntu Server 20.04 ISO under the *Storage* sub-menu, ensure *Adapter 1* is the only adapter enabled under the *Network* sub-menu, and that its attached to the *Host-only Adapter*. Windows systems should display the Name *VirtualBox Host-Only Ethernet Adapter*, while Linux/MacOS systems should display *vboxnet0*. Record the MAC address, and ensure the *Cable Connected* checkbox is checked. Finally, make a static DHCP mapping on the LAN interface of the pfSense DHCP server for the SIEM VM, reserving the IP address 172.16.1.3.

IPS VM:

Storage: Mount the Ubuntu Server ISO to the IPS VM's virtual CD/DVD drive just like with the SIEM VM.

Network: Ensure the *Enable Network Adapter* checkbox is checked on the *Adapter* tabs 1 through 3.

Adapter 1:

Attached To: *Host-Only Adapter*

Name: *VirtualBox Host-Only Ethernet Adapter* (Windows), *vboxnet0* (Linux/macOS)

Click *Advanced* button, and record the contents of the *MAC Address* field, and Confirm the *Cable Connected* checkbox is checked. Use this MAC address to create a static DHCP mapping on *LAN* interface of the pfSense VM, and assign it the IP address 172.16.1.4. Again, chapter 14, [section 14.3.4.1](#) (pp. 690-692) can help provide instructions on how to perform this task, if necessary.

Adapter 2:

Attached To: Internal Network

Name: intnet

Click *Advanced* button, record the contents of the *MAC Address* field, and Confirm the *Cable Connected* checkbox is checked. **The last and most important thing to enable for this interface is to set the drop-down labeled Promiscuous Mode. Set it to Allow All.**

Adapter 3:

Attached To: Internal Network

Name: intnet1

Just like *Adapter 2* above, record the *MAC Address* field, make sure *Cable Connected* is checked, and **make absolutely sure Promiscuous Mode is set to Allow All.**

The image shows a series of screenshots from the pfSense configuration interface. At the top left, a sidebar menu has 'Storage', 'Audio', and 'Network' items. A red arrow points from 'Storage' to a 'Storage' configuration window where 'Controller: IDE' and 'ubuntu-20.04-live-server-amd6...' are highlighted. Another red arrow points from 'Network' to three network adapter configuration windows. The first window (Adapter 2) shows 'Attached to: Internal Network', 'Name: intnet', 'Adapter Type: Intel PRO/1000 MT Desktop (82540EM)', 'Promiscuous Mode: Allow All', and 'MAC Address: 080027C919E8'. The second window (Adapter 3) shows 'Attached to: Internal Network', 'Name: intnet1', 'Adapter Type: Intel PRO/1000 MT Desktop (82540EM)', 'Promiscuous Mode: Allow All', and 'MAC Address: 0800273FDA0B'. The third window (Adapter 1) shows 'Attached to: Host-only Adapter', 'Name: VirtualBox Host-Only Ethernet Adapter', 'Adapter Type: Intel PRO/1000 MT Desktop (82540EM)', 'Promiscuous Mode: Deny', and 'MAC Address: 080027EB2DE9'. A red arrow points from the MAC address '080027EB2DE9' in the third window to a table below. The table is titled 'DHCP Static Mappings for this Interface' and has columns for 'Static ARP', 'MAC address', 'IP address', 'Hostname', and 'Description'. The second row of the table is highlighted with a red box, showing '0800:27:eb:2d:e9' mapped to IP '172.16.1.4' for hostname 'IPS'.

Services / DHCP Server / LAN

DHCP Static Mappings for this Interface				
Static ARP	MAC address	IP address	Hostname	Description
	08:00:27:b8:ba:24	172.16.1.3	SIEM	DHCP mapping for SIEM VM
	08:00:27:eb:2d:e9	172.16.1.4	IPS	DHCP mapping for IPS VM

10-35: Mount the Ubuntu Server ISO under the *Storage* sub-menu, and ensure *Adapter 1*, *Adapter 2*, and *Adapter 3* are all enabled. *Adapter 1* should be attached to the *Host-only Adapter*, just like the SIEM VM in *fig. 10-31*. Record the MAC address, and make sure the *Cable Connected* checkbox is checked. Use this information make a static DHCP mapping on the LAN interface of the pfSense DHCP server for the IPS VM, reserving the IP address 172.16.1.4. *Adapter 2* and *Adapter 3* should both have the *Attached to* drop-down set to *Internal Network*. In the *Name* input box, enter *intnet* for *Adapter 2*, and *intnet1* for *Adapter 3*. Under *Advanced*, record the MAC addresses for Adapters 2 and 3, and ensure the *Cable Connected* checkbox is checked. Finally, **for adapters 2 and 3, set the Promiscuous Mode drop-down to Allow All. This is extremely important for later chapters!**

Kali VM:

Storage: Mount the Kali Linux ISO to the IPS VM's virtual CD/DVD drive.

Network: Ensure the *Enable Network Adapter* checkbox is checked on the *Adapter 1*.

Adapter 1:

Attached To: *Internal Network*

Name: *intnet*

Click *Advanced* button, record the contents of the *MAC Address* field, and Confirm the *Cable Connected* checkbox is checked. Use this MAC address to create a static DHCP mapping on *OPT1* interface of the pfSense VM, and assign it the IP address 172.16.2.3. Just like with the SIEM and IPS VMs, chapter 14, [section 14.3.4.1](#) (pp. 690-692) provides guidance on how to perform this task.

Storage Devices

Controller: IDE

kali-linux-2020.2-installer-amd6...

Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

Enable Network Adapter

Attached to: Internal Network

Name: intnet

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08002769E90C

Cable Connected

Services / DHCP Server / OPT1

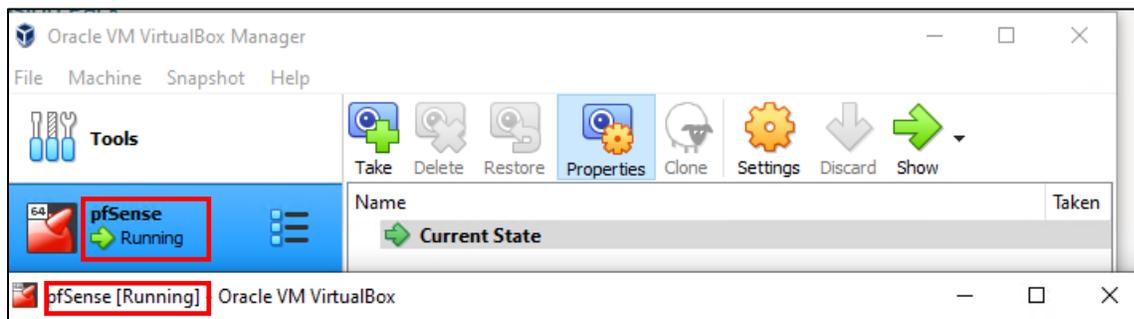
DHCP Static Mappings for this Interface				
Static ARP	MAC address	IP address	Hostname	Description
	08:00:27:69:e9:0c	172.16.2.2	kali	DHCP mapping for kali VM

10-36: Mount the Kali Linux ISO under the *Storage* sub-menu, ensure *Adapter 1* is the only adapter enabled under the *Network* sub-menu, and make sure its attached to *Internal Network*, with the *Name* input box set to *intnet*. Record the MAC address, and make sure the *Cable Connected* checkbox is checked. Finally, make a static DHCP mapping on the *OPT1* interface of the pfSense DHCP server for the kali VM, reserving the IP address 172.16.2.2.

10.7.2 Operating System Installation

In this section, students will learn how to install the operating system for the SIEM, IPS, and Kali virtual machines. Both the SIEM and IPS VMs will have Ubuntu Server 20.04 installed as their operating system, while the Kali VM will have the latest version of the Kali Linux distribution installed. The installation instructions will differ for each virtual machine, so please pay attention.

As a general reminder, students should make sure that the pfSense VM is running, and that they have completed chapter 14 to ensure pfSense is ready to support the rest of the lab environment. Without the pfSense VM, none of the virtual machines will have internet access. That may result in the operating system installers failing in different ways. To confirm the pfSense is running, students can click on the pfSense virtual console. The name of the virtual console window should read *pfSense [Running] – Oracle VM VirtualBox*. Alternatively, the *VirtualBox Manager* window will show the pfSense VM with a status of *Running*.



10-37: The *VirtualBox Manager* window can be used to confirm that the pfSense VM is running. Likewise, the title of the pfSense virtual console window will also reflect the current operating state (e.g. *[Running]*).

10.7.2.1 Installing Ubuntu on the SIEM VM

To get started, highlight the SIEM VM in the VirtualBox Manager window, and click the big green *Start* button to power on the SIEM virtual machine, and open up the SIEM VM's virtual console. Please note, that if this is your first time starting the virtual machine, a window titled *Select start-up disk* will appear. Click *Cancel* to continue (we already handled this in section 10.7.1). The virtual machine will begin booting off the Ubuntu Server ISO. The first screen will ask you to confirm the language you wish to use. The default language should be *English*, so hit the enter key on your keyboard to continue.

Depending on when students downloaded their copy of the Ubuntu Server ISO, and how frequently the ISO is updated, a screen may appear titled *Installer update available*. This screen provides users with the option to download the latest version of the Ubuntu installation wizard, called Subiquity. Highlight *Update to the new installer* using the arrow keys on the keyboard, then hit enter.

Note: If for some reason downloading the latest installer fails, there's a good chance that there are network problems with the lab environment elsewhere, and that there is troubleshooting to do. Students are welcome to select the *Continue without updating* option, but keep this in mind if the installer misbehaves or fails later. Check to see if the hypervisor host has internet connectivity, double check the firewall rules on the pfSense virtual machine, network settings, physical cabling, etc.

The next screen asks users to confirm their keyboard configuration. The default settings for both the *Layout* and *Variant* settings are *English (US)*. If students are not using a standard US-English keyboard, use the arrow keys to highlight the *Identify keyboard* option, then hit enter. Otherwise, highlight *Done* on the bottom of the screen, and hit enter.

Next up, is the *Network connections* screen. If students followed the instructions in [section 10.7.1](#) (pp. 249-254) *fig. 10-34*, and created a static DHCP allocation for the SIEM VM, a single network adapter should populate this page. The network adapter (named `enp0s3` in my case) should automatically be assigned the IP address 172.16.1.3. Below the IP address in light grey text is the MAC address of the network adapter that the Ubuntu installer detected. This should be the same MAC address of adapter 1 of the SIEM VM. If the correct IP address was assigned, students can hit the enter key to continue (The *Done* option should be highlighted by default). Otherwise, see the sidebar conversation below, *What Reservation?* for some troubleshooting tips.

What Reservation?

If for some reason the network adapter was assigned any other IP address other than 172.16.1.3, Refer back to [section 10.7.1](#), pp. 249-254. Check the MAC address of adapter 1 of the SIEM VM. Compare that MAC address to the MAC address used to create a static DHCP mapping on the *LAN* interface of the pfSense VM. Compare that to the MAC address displayed on the *Network connections* screen of the Ubuntu installer. ***They should all be identical!*** (missing colon [:] symbols notwithstanding). If there are any errors, correct them and reset the SIEM VM, to restart the Ubuntu installer until the pfSense DHCP assigns the network adapter the correct IP address.

MAC Address: 080027B8BA24	08:00:27:b8:ba:24	172.16.1.3	DHCPv4 172.16.1.3/24 08:00:27:b8:ba:24 / In
---------------------------	-------------------	------------	--

10-38: If the SIEM VM failed to get the correct IP address, check the network settings of *Adapter 1* in VirtualBox, the MAC address used to create a static DHCP mapping on the *LAN* interface on the pfSense WebConfigurator, and compare those to the MAC address displayed by the *Network connections* screen of the Ubuntu installer.

The *Configure proxy* screen appears. Use the up arrow key to highlight the text box labeled *Proxy address* and enter `http://172.16.1.1:3128`. Recall from Chapter 14, this is the IP address and port for the Squid proxy on the *LAN* interface of the pfSense VM. Use the arrow keys to highlight *Done*, and hit enter to continue.

The next screen, labeled *Configure Ubuntu archive mirror* will appear. This is another one of those situations where students will know whether or not they need to change this setting Unless the lab environment is in an enterprise network and the network team happens to be operating their own software archive mirror, accept the default setting (in my case, the default mirror address was <http://us.archive.ubuntu.com/ubuntu>). With *Done* highlighted, hit enter to continue.

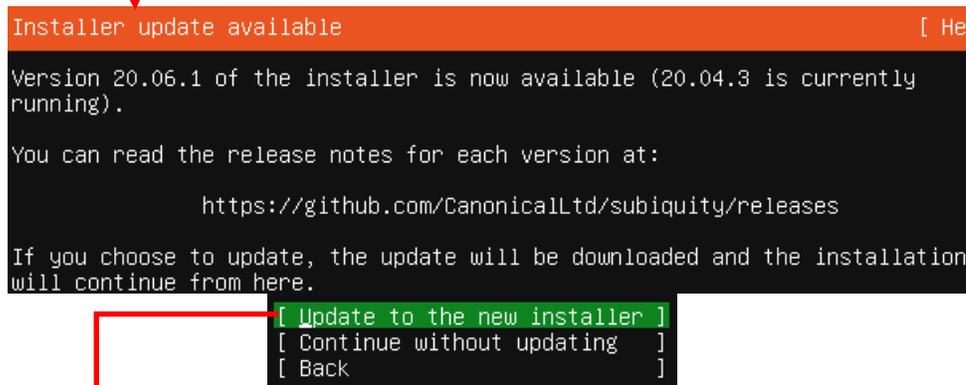
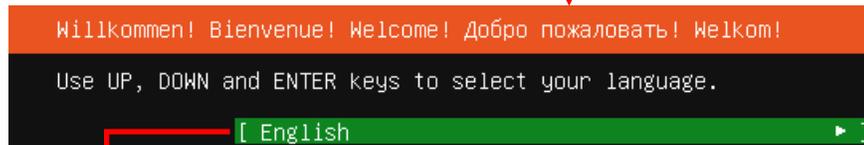
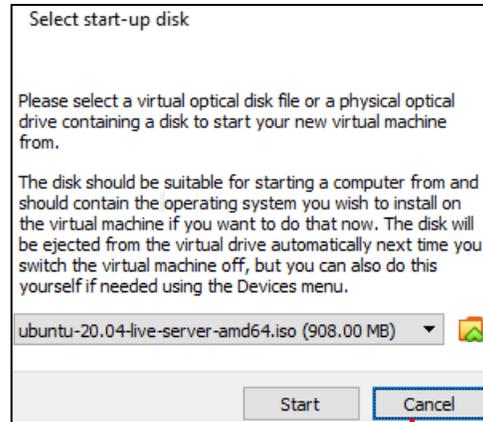
The *Guided storage configuration* screen appears next. Accept the default settings and let the Ubuntu installer format the entire disk. Use the arrow keys to highlight *Done*, and hit enter to continue. The next screen, titled *Storage configuration*, shows you how the installer is going format the hard drive, and what partitions are going to be where in a large list labeled *FILE SYSTEM SUMMARY*. By default, *Done* should already be highlighted on this screen. If not, use the arrow keys to highlight it and hit enter to continue. A pop-up labeled *Confirm destructive action* appears. This screen informs the user that any data on the disk will be lost as a result of formatting and partitioning the disk. Since there is no data on the virtual hard disk yet, highlight *Continue*, and hit the enter key to proceed.

Next is the *Profile setup* screen. There are five input boxes on this screen. Ubuntu asks the user for their name, the server's name, a username (that will be used to log in to the server later), the password for that username, followed by an input box asking the user to repeat the password. Students may enter any name, username, or password they would like, but it is recommended to both set the server name to *siem*, as well as to save the username and password combination to a password manager. Once finished, use the arrow keys to highlight *Done*, then hit enter to continue.

The *SSH Setup* screen appears and asks users if they would like to install the OpenSSH server package by default. By default, the prompt should be between two brackets next to the text *Install OpenSSH server*. Hit the spacebar to leave an 'X' between the brackets. Afterwards, use the arrow keys to highlight *Done* and hit enter to advance.

The next screen is labeled *Featured Server Snaps*. The latest versions of Ubuntu use an additional software manager called 'snap' to deliver software packages. Use either the arrow keys or the tab key to highlight *Done* and hit enter to proceed.

We reach a new screen labeled *Install complete!* At this point, students have made all of the necessary decisions for the ubuntu installer to proceed, and handle all of the installation tasks at once. Once completed, the installer will grant students the option to reboot the system. However, instead of using the installer's reboot function to reboot, close the SIEM virtual machine's virtual console. A small window will appear asking what you would like it to do. Select the option *Power off the machine*, and click *OK*.



Continued in *fig. 10-40*

10-39: If prompted by the *Select start-up disk* window, students should have already assigned the Ubuntu Server 20.04 installation ISO, so click *Start* to proceed. Students then select the language of the installer, the installer checks for updates for itself, then asks the user to set the keyboard layout and variant language.

Continued from *fig. 10-39*

```
Network connections [ Help ]
Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.

NAME      TYPE  NOTES
[ enp0s3  eth  -           ▶ ]
  DHCPv4  172.16.1.3/24
  08:00:27:b8:ba:24 / Intel Corporation / 82540EM Gigabit Ethernet Controller
  (PRO/1000 MT Desktop Adapter)
```

```
[ Done ]
[ Back ]
```

```
Configure proxy [ Help ]
If this system requires a proxy to connect to the internet, enter its details
here.

Proxy address: http://172.16.1.1:3128
If you need to use a HTTP proxy to access the outside world,
enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of
"http://[[user] [:pass]@]host[:port]/".
```

```
[ Done ]
[ Back ]
```

```
Configure Ubuntu archive mirror [ Help ]
If you use an alternative mirror for Ubuntu, enter its details here.

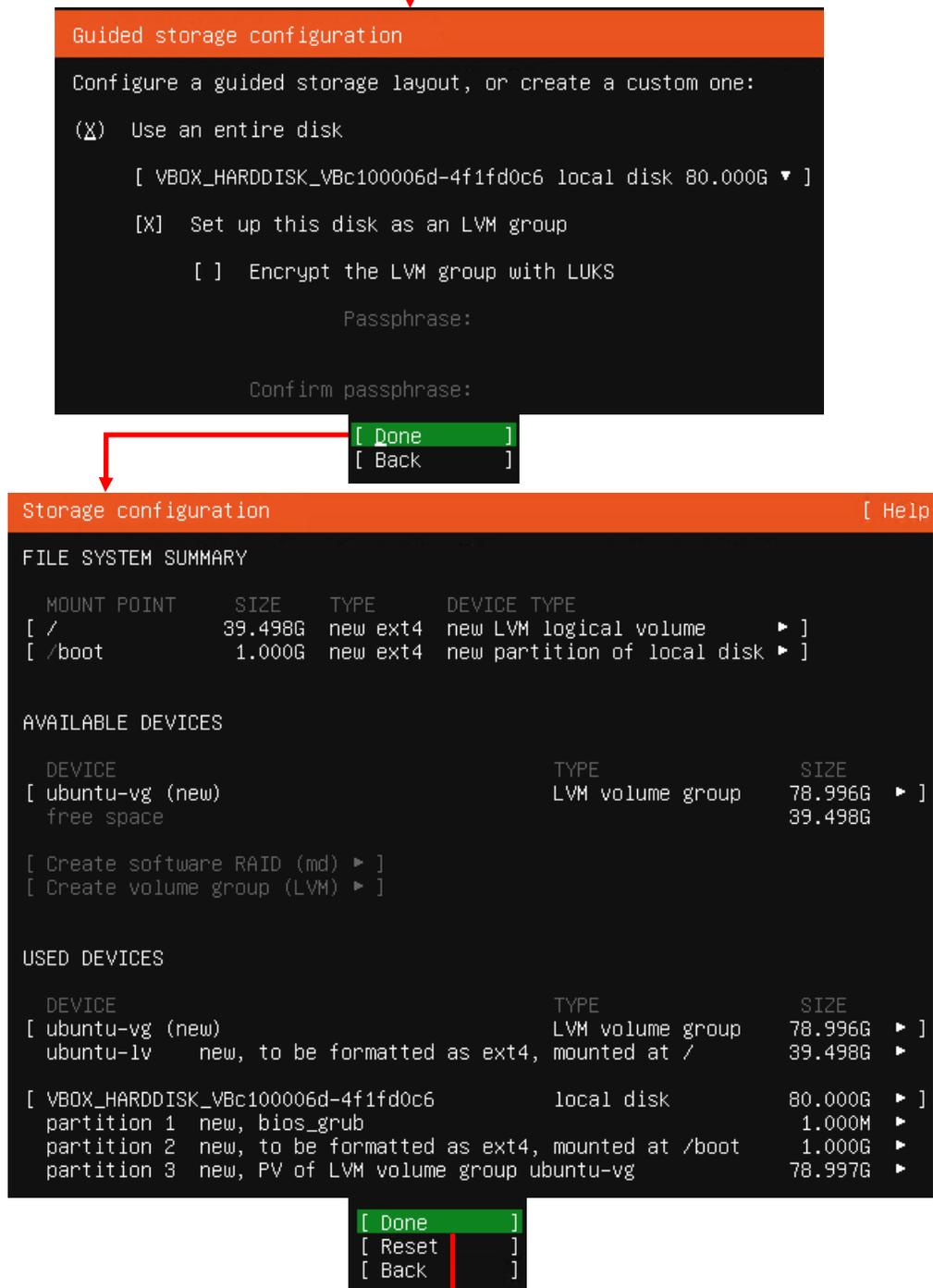
Mirror address: http://us.archive.ubuntu.com/ubuntu
You may provide an archive mirror that will be used instead of
the default.
```

```
[ Done ]
[ Back ]
```

Continued to *fig. 10-41*

10-40: The next stages of the Ubuntu Server 20.04 installer. In these screens, students can confirm whether or not the static DHCP mapping for the SIEM VM is working correctly, configure the system to use the Squid proxy service configured on the pfSense VM, and confirm software archive mirror they would like to use.

Continued from *fig. 10-40*



Continued to *fig. 10-42*

10-41: These screens are used to configure the storage settings for the operating system. Students will be using the default storage settings for the SIEM VM.

Continued from *fig. 10-41*

Confirm destructive action

Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the installation has started.

Are you sure you want to continue?

[No]
[Continue]

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: ayy

Your server's name: siem
The name it uses when it talks to other computers.

Pick a username: ayy

Choose a password: *****

Confirm your password: *****

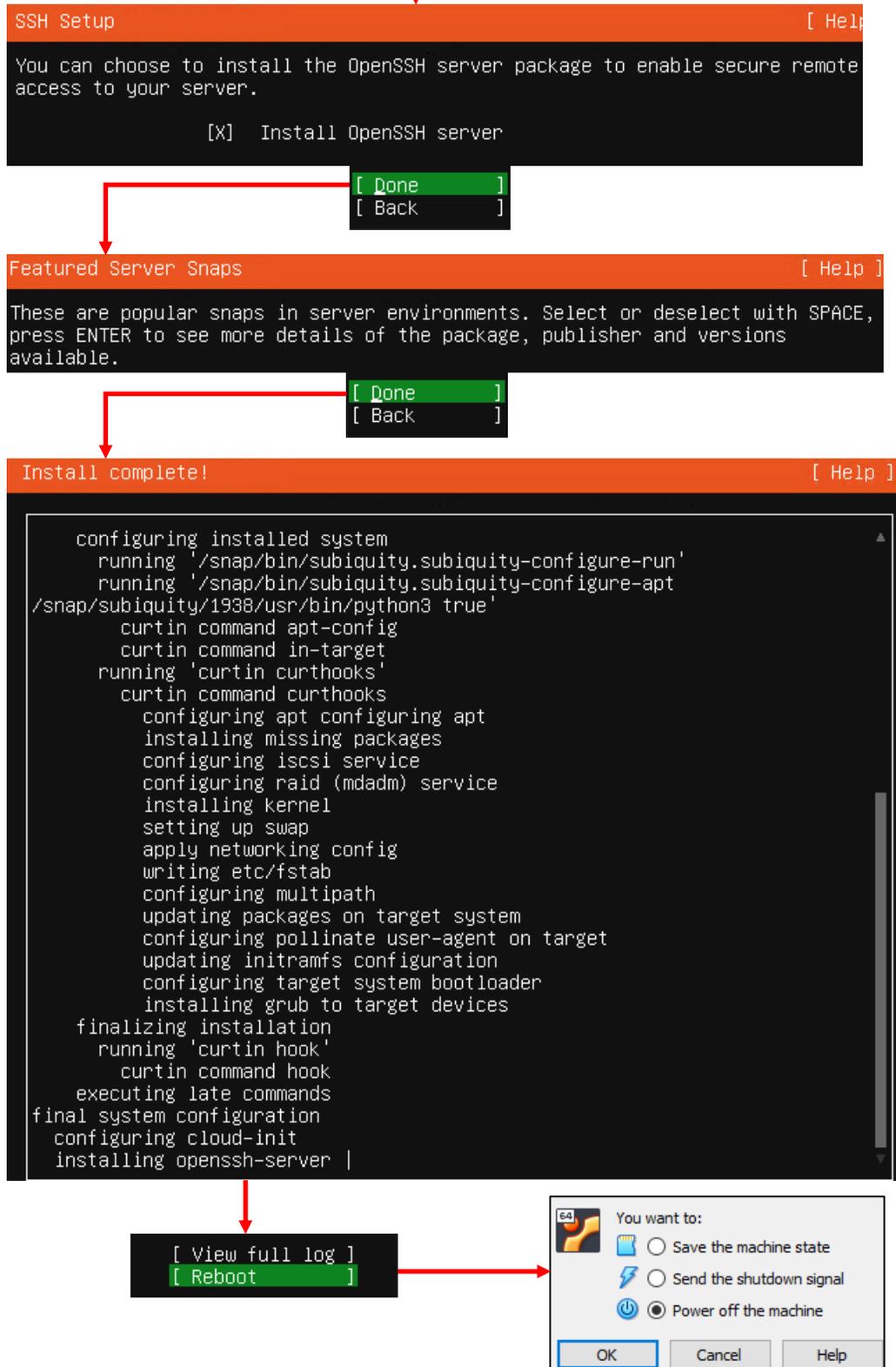
[Done]

Title:	SIEM VM
Username:	ayy
Password:	*****
URL:	172.16.1.3
<input type="checkbox"/> Expires:	7/21/2020 12:11 PM
<input checked="" type="checkbox"/> Notes:	User account credentials for the SIEM VM.

Continued to *fig. 10-43*

10-42: After confirming the storage configuration settings, users are prompted to name their server, and create a user account. It's recommended to store the username and password for the SIEM VM in a password manager.

Continued from *fig. 10-42*



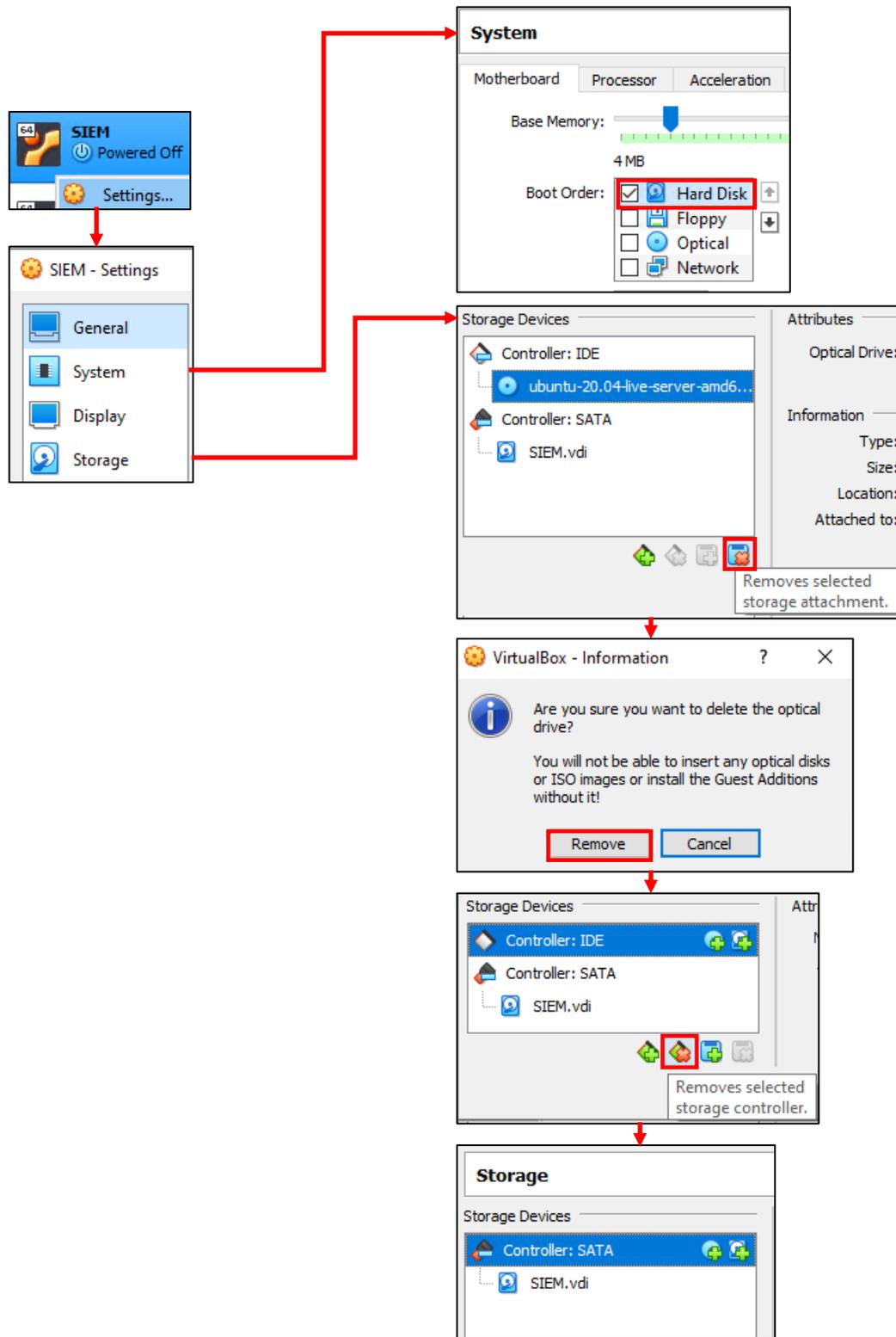
10-43: The final stages of the Ubuntu Server 20.04 installer. Select the option to install OpenSSH server, then decline to install any server snaps. Finally, once the installer grants you the option to reboot, ***close the virtual console and power off the virtual machine.***

10.7.2.2 Additional Virtual Machine Settings – SIEM VM

Now that the operating system installation is complete, we have a handful of virtual machine settings to change, just like with the pfSense VM in *section 10.6.4*. In the *VirtualBox Manager* window, right click on the SIEM VM in the virtual machine inventory list, select *Settings* and make the following changes:

In the *System* menu, under the *Motherboard* tab, Modify the *Boot Order* list. Uncheck the *Optical* and *Floppy* checkboxes, then highlight *Hard Disk*, and use the small arrow icon to make it the very first item at the top of the list.

In the *Storage* menu, in the *Storage Devices* window, click on the blue icon that looks like a CD to highlight it. This is the Virtual CD/DVD drive we used to install Ubuntu to the SIEM VM. At the bottom of the window are four small icons – Hover over the square with the x over it and a box will pop up that reads *Removes selected storage attachment*. click on square to remove the virtual CD/DVD drive. A pop-up box appears asking if you are sure you want to remove the optical drive. Click the *Remove* button to continue. The *Storage Devices* listing updates to reflect that we no longer have a virtual CD/DVD drive. Next, click on the item labeled *Controller:IDE* to highlight it. Hover over the small blue diamond with an x over it at the bottom of the window and a box appears that reads *Removes selected storage controller*. Clicking on the icon immediately removes the IDE controller with no additional dialogue box immediately. When you are finished, there should only be two items in the *Storage Devices* window: *Controller:SATA*, and the virtual disk labeled *SIEM.vdi*. Click *OK* in the bottom right corner of the *SIEM-Settings* window to apply both the *System* and *Storage* menu settings.



10-44: Open the *Settings* menu of the SIEM VM, and edit the boot order, remove the virtual CD/DVD ROM drive, then remove the IDE controller as well. Once finished, click *OK* to exit the settings menu and confirm these changes.

10.7.2.3 Booting the SIEM VM for the first time

After changing the SIEM VM's settings, click the start button in the *VirtualBox Manager* to start the VM back up and bring up its virtual console. After a moment or two, students will be greeted with a login prompt labeled `siem login`. Enter the username configured during the installer, followed by its password to log in.

Some students may not be familiar with command-line applications, and that's okay. This is only a quick login to make sure network connectivity is working. Please type in the following commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The purpose of the `ip` command above is to display all of the network interfaces on the system. We pass this command the `-br` option for brief output, followed by the letter `a` to indicate we're interested in seeing the IP addresses on our system. Users could replace `a` with `address` or `addr` and the `ip` command would interpret it the same. We're using this command to serve as a secondary confirmation that the SIEM VM was successfully assigned the IP address `172.16.1.3`, as displayed in *fig.10-45*. Students may notice a second interface on the system designated `lo`. This is a "loopback" network interface and can be ignored.

The `nslookup` command is to confirm that the SIEM VM is able to resolve hostnames using DNS. The output from the command should be similar to what is presented in *fig. 10-45*. Finally, that brings us to the `curl` command. This command is to confirm connectivity to the internet over port 443, HTTPS. The `-I` option in the command tells `curl` to only return the HTTP headers from the web server being contacted. Once again, the output of this command should be fairly similar to what is presented in *fig. 10-45*.

```

Ubuntu 20.04 LTS siem tty1

siem login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ nslookup www.google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:806::2004

ayy@siem:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 21 Jul 2020 20:10:38 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 21 Jul 2020 20:10:38 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-21-20; expires=Thu, 20-Aug-2020 20:10:38 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=1AAB5nk21PEgo8rGiFr-9PxEuTIY0Nx2tMMi-EmACtdRnP1PkB0xoosGu9FjWzbLyW0TGOHKUj6kLonn4Rr-yu-MIc8itYAI507X2VkJk1WkOUK30rSk6Fd0ce6_Batt9PQ4YI7-CoRGf381n74m078YmTAAtz1XJf8-WM; expires=Wed, 20-Jan-2021 20:10:38 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@siem:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8      ::1/128
enp0s3            UP                172.16.1.3/24   fe80::a00:27ff:feb8:ba24/64

```

10-45: After logging in to the SIEM virtual machine, students will need to run a series of network troubleshooting commands. These commands are to confirm that the SIEM VM has the correct IP address configured (`ip -br a`), can resolve hostnames via DNS (`nslookup www.google.com`), and has connectivity over HTTPS (`curl -I https://www.google.com`).

Before logging out of the SIEM virtual machine, there are three more commands to run, but before we can run them, we will need to become the root user. Enter the following command:

```
sudo su -
```

When prompted, enter the password for the user you created. If successful, you will be logged in as the root user on the SIEM virtual machine. The root user, sometimes referred to as the super user, is a special account that has complete authority over the system. Additionally, root has access to special administrative commands that normal users are not allowed to use. As the root user, **run the following commands in this exact order:**

```
apt-get update
apt-get -y dist-upgrade
init 6
```

Ubuntu is based off of the Debian Linux distribution. Because of this, it uses a package manager called apt (in addition to the snap package manager mentioned earlier). The two apt-get commands, apt-get update then apt-get -y dist-upgrade tell Ubuntu to reach out to the software archive mirror and get an updated list of software packages, then if any packages installed on the system need to be updated, download and install the updates immediately. This set of commands also confirms that the Squid proxy server on the pfSense VM is working properly, and proxying all of the HTTP requests from the SIEM VM. The final command, init 6, tells the system to reboot immediately. As an alternative, users can also run the command reboot instead.

```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (524 kB/s)
Reading package lists... Done
root@siem:~# apt-get -y dist-upgrade
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@siem:~# init 6_
```

10-46: the command `sudo su -` allows students to become the root user on the SIEM VM. We then use root's permissions to ensure all of the software packages on the system are up to date (`apt-get update`, followed by `apt-get -y dist-upgrade`), then immediately reboot the system (`init 6`, or optionally `reboot`). Be aware that the apt-get commands may take a little bit of time to finish, based on the number of updates available and speed of the internet connection.

Help! My apt-get commands are failing!

If you're experiencing problems with the apt-get commands failing to complete, there's a very good chance that the apt package manager is not properly configured to use the SQUID HTTP proxy we installed on the pfSense VM, or that the SQUID proxy service on the pfSense VM may be misconfigured. If students entered the wrong information during the operation system installation (e.g. on the Configure Proxy screen), then the apt package manager will not work properly.

Here are some troubleshooting steps to think about:

On the SIEM VM, run the command:

```
cat /etc/apt/apt.conf.d/90curtin-aptproxy
```

This command will read the contents of the file `/etc/apt/apt.conf.d/90curtin-aptproxy` and display its contents on the screen. The file should read something like this:

```
Acquire::http::Proxy "http://172.16.1.1:3128";  
Acquire::https::Proxy "http://172.16.1.1:3128";
```

If this file does not exist, or has any content that is in any way different from the lines above, **run the following three commands exactly as displayed, and in this exact order:**

```
sudo su -  
echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-  
aptproxy  
echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >>  
/etc/apt/apt.conf.d/90curtin-aptproxy
```

This series of commands requires root access, so the first thing we do is use `sudo su -` to become the root user. The next two commands delete the current `90-curtin-aptproxy` file if it exists, then overwrites it with the two correct entries that should exist in the file. After running these commands, run `cat /etc/apt/apt.conf.d/90curtin-aptproxy` once more, and confirm that the output matches the correct output listed above. After confirming that the configuration file has been recreated correctly, try running the apt-get commands once more. If they continue to fail, then continue the troubleshooting process. Assuming that the network connectivity check commands were successful (e.g. `nslookup` and `curl`), think about the following:

- Is the SQUID proxy service installed on pfSense?
- Is there a firewall rule on the LAN interface to allow access to the proxy service? (allow traffic to IP address 172.16.1.1 port 3128 TCP from network 172.16.1.0/24)
- Is the option *Resolve DNS IPv4 First* checked on the SQUID proxy service?

These are all configurations covered in chapter 14, and should have already been specified. Double check that they have been configured correctly, then try updating the SIEM VM again.

```

ayy@siem:~$ 1 sudo su -
root@siem:~# 2 echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 3 echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >> /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 4 cat /etc/apt/apt.conf.d/90curtin-aptproxy
Acquire::http::Proxy "http://172.16.1.1:3128";
Acquire::https::Proxy "http://172.16.1.1:3128";
root@siem:~# 5 apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [332 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [8,780 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [5,404 B]
Fetched 826 kB in 1s (972 kB/s)
Reading package lists... Done
root@siem:~# 6 apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@siem:~#

```

10-47: This illustration demonstrates how to fix or modify the `/etc/apt/apt.conf.d/90curtin-aptproxy` file, in the event that student find that there is a problem with the file. First utilize `sudo su -` (1) to become the root user. Then use the two `echo` commands (2, 3) to write the correct configuration data so that `apt` knows how and where to access the squid proxy configured on the pfSense VM. Utilize the `cat` (4) command to confirm that the configuration file is properly configured. Finally, run `apt-get update` (5) and `apt-get -y dist-upgrade` (6) to check for the latest updates and download them.

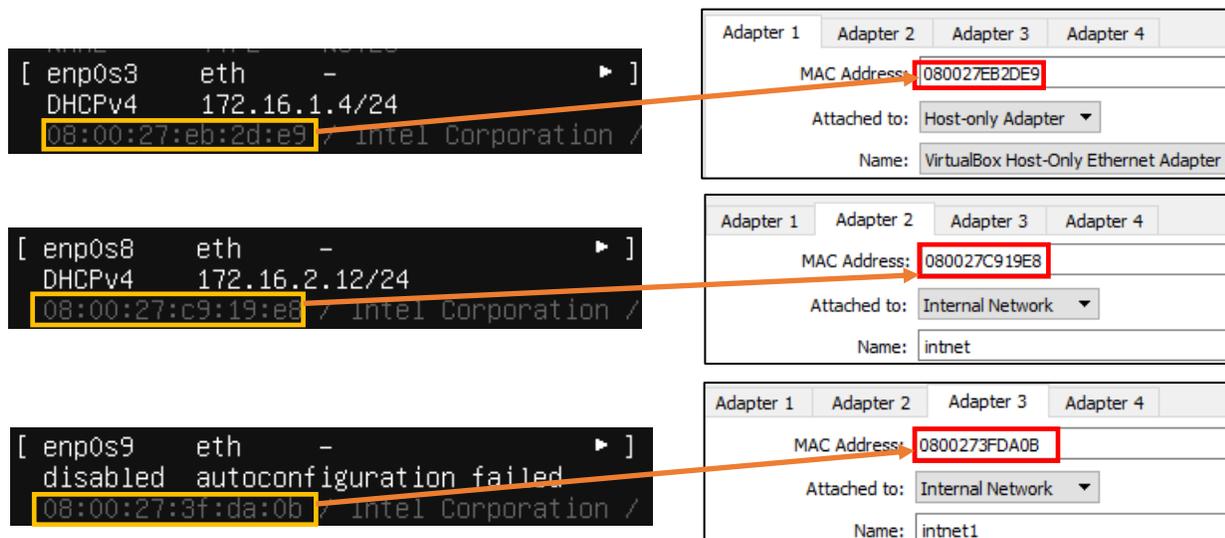
10.7.2.4 Installing Ubuntu on the IPS VM

Now that Ubuntu Server has been installed on the SIEM VM, network connectivity has been checked, and updates have been applied, next up is the IPS VM. The process for installing Ubuntu Server on the IPS virtual machine is practically identical. The process will be summarized below, with major differences to be aware of explained further in-depth.

- Start the IPS VM, and connect to its virtual console
- Select English as your language (or your preferred language)
- If there are any updates to Subiquity, select the option, *Update to the new installer*
- Select *English (US)* (or your preferred language) as the keyboard *Layout* and *Variant*

The *Network connections* screen will be a little bit different than it was on the SIEM VM, because the IPS virtual machine has three network interfaces. Recall in section 10.6.5, comparing and contrasting the MAC addresses of adapter 1, adapter 2, and adapter 3 for the pfSense VM, and using that information to map the name of the network interface in pfSense (e.g., adapter 1 → em0 → WAN, adapter 2 → em1 → LAN, adapter 3 → em2 → OPT1).

Students will need to perform a similar exercise for the IPS virtual machine on the *Network connections* screen. In light grey text underneath the name of each network interface is the MAC address for that interface. Cross-reference the MAC address and interface name on the screen with the MAC address of adapters 1-3 recorded earlier. See *fig. 10-48* below for an example, based on the MAC addresses of my IPS virtual machine.



10-48: Ubuntu has assigned our three virtual adapters the interface names enp0s3, enp0s8, and enp0s9. Below each interface name is a MAC address. By cross referencing that MAC address to *Adapter 1*, *Adapter 2*, and *Adapter 3*, we can determine which interface name maps to which virtual adapter, and confirm which network segment the interfaces are attached to. For instance, enp0s3 is *adapter 1*, attached to the management/LAN network. enp0s8 is *adapter 2* is attached to IPS 1/intnet, while enp0s9 is *adapter 3*, attached to IPS 2/intnet1.

Now that students are aware of which interface corresponds to which network segment, the next step is ensuring that the interface connected to the Management network (e.g., the LAN network in pfSense, or the Host-only Network in VirtualBox) is the only interface that has an IP address assigned. In [section 10.7.1 \(pp. 249-254\)](#), [fig. 10-35](#), students created a static DHCP reservation for *adapter 1* of the IPS VM, and assigned it the IP address 172.16.1.4. In [fig. 10-48](#), the interface `enp0s3` has the IP address 172.16.1.4. I was able to confirm that this interface maps to *adapter 1*, attached to the LAN network. This confirms I created the static DHCP allocation correctly on the pfSense WebConfigurator. If the network adapter attached to the LAN network does not have the correct IP address, there is a good chance that the static DHCP mapping for the IPS virtual machine is incorrect. Take a look at the side bar discussion, *Reservation for One*, for some troubleshooting recommendations.

Reservation for One

If you're here, that means that the network interface attached to the LAN/Management network didn't get the IP address 172.16.1.4. Similar to the *What Reservation?* Sidebar discussion for the SIEM VM, you'll want to check a few things:

- Check the MAC address field of *Adapter 1* in the IPS virtual machine settings
Visit *Services > DHCP Server* and Check the Static DHCP Mappings of the LAN interface, particularly, the entry for the IPS VM
- Compare the MAC address of the previous two locations with the MAC addresses presented on the *Network Connections* screen. You should already know which interface name maps to VirtualBox *adapter 1*. In my case this was the interface `enp0s3`
- Make any corrections to the static DHCP allocation, then restart the IPS VM. Make your way back to the Network connections screen, and confirm that the correct interface was assigned the correct IP address.

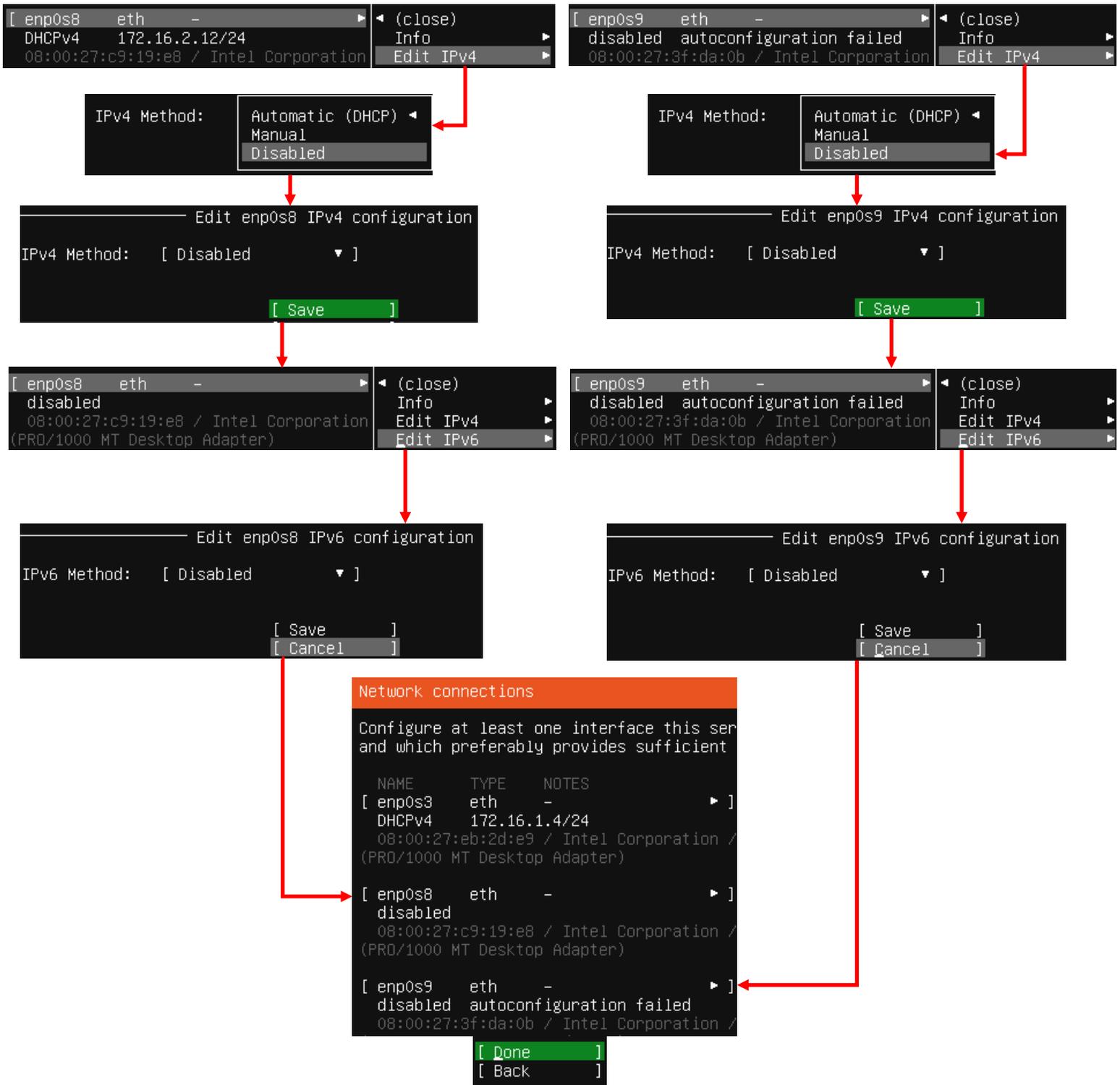
Adapter 1	Adapter 2	Adapter 3	08:00:27:eb:2d:e9	172.16.1.4	[enp0s3 eth -]
MAC Address: 080027EB2DE9					DHCPv4 172.16.1.4/24
					08:00:27:eb:2d:e9 / Intel Corporation /

10-49: Just like with the SIEM VM, compare the MAC address of *Adapter 1* to the MAC address you entered for the static DHCP allocation on pfSense, to the MAC address/interface name presented in the Ubuntu installer. All three MAC addresses should be identical. If they aren't correct the DHCP mapping on pfSense, restart the IPS VM, and confirm it gets the correct IP address.

The final step on the *Network connections* screen is to disable the remaining network interfaces. The interfaces connected to *intnet* and *intnet1* (e.g., the interfaces *enp0s8*, and *enp0s9*) should never receive an IP address. The lab environment, and IPS software we'll be using does not require these interfaces to have IP addresses, so we want to take advantage of that. Students may have notice that the interface connected to *intnet1* (*enp0s9*) doesn't have an IP address assigned, instead displaying the status: *disabled autoconfiguration failed*. Disregard this error message and follow the instructions below. Substitute the interface names *enp0s8* and *enp0s9* as necessary:

- Using the arrow keys, Highlight one of the other remaining interfaces. In my case, I chose to highlight *enp0s8*. Hit enter, and a dialogue box pops up.
 - Highlight the option *Edit IPv4*, and hit enter.
 - A new dialogue box appears titled *Edit enp0s8 IPv4 configuration*, with a single drop-down option highlighted, titled *IPv4 Method*. Hit enter again, and a list of choices appear. Use the arrow keys to select the option *Disabled*, and hit enter.
 - Use the arrow keys to highlight the option *Save*, and hit enter.
- Optional: Repeat the process again, only this time, Select *Edit IPv6*, and verify that it is set to *Disabled* (This is usually the default setting).
 - When finished, exit the *Edit enp0s8 IPv6 configuration* dialogue box.
- Repeat this process for the final interface. In my case, *enp0s9*. *Disable* the IPv4 Configuration (in my case, it was already set to *Disabled*) and confirm that the IPv6 configuration is already *Disabled*.

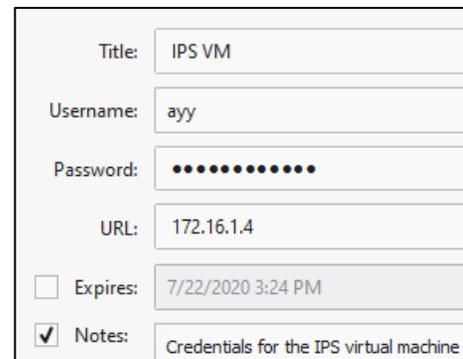
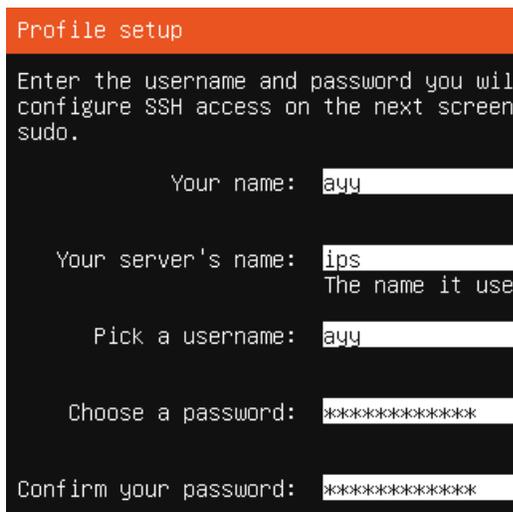
The end result should be one interface with the IP address 172.16.1.4, and two disabled network interfaces. Students can refer to *fig. 10-50* for assistance. When finished, use the arrow keys to highlight *Done*, and hit enter to continue.



10-50: enp0s3 is the interface attached to the LAN network, and should be the only interface with an IP address. ***Disable the other interfaces. They should never be assigned an IP address.***

The rest of the installation process for the IPS VM should be identical to the SIEM VM:

- On the *Configure proxy* screen, set the *Proxy address* to `http://172.16.1.1:3128`
Accept the default archive mirror (or an alternative, if required) on the *Configure Ubuntu archive mirror* screen
- Accept the default settings on the *Guided storage configuration*, and *Storage configuration* screens. Select *Continue* on the *Confirm destructive action* dialogue pop-up
- Fill out the *Profile setup* screen, ensuring that the *Your server's name* input box is set to *ips*. Remember to document the username and password you create and store it in your preferred password manager

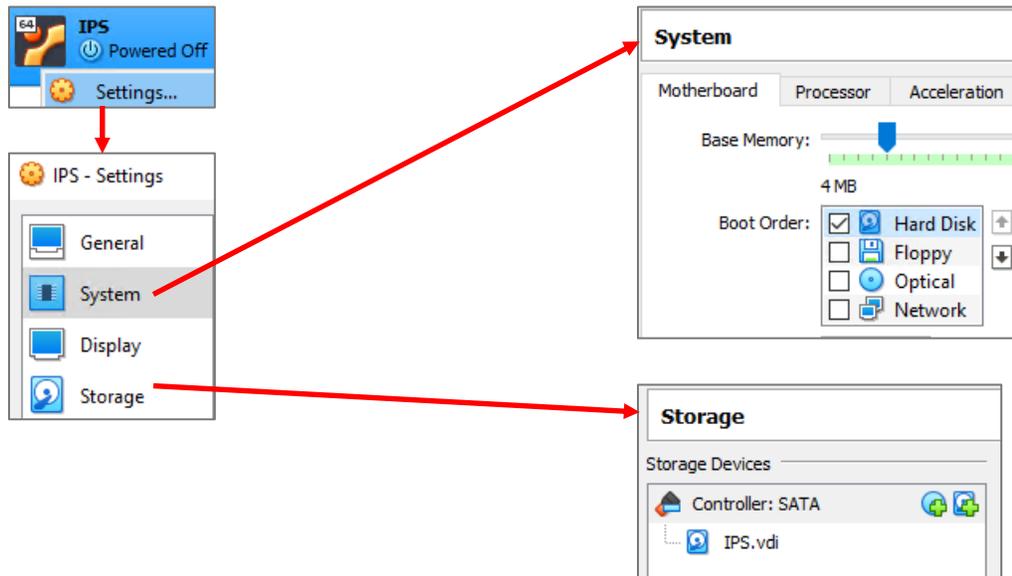


10-51: The *Profile setup* screen for the IPS virtual machine is, quite literally, the only other screen aside from the *Network connections* screen that differs from the installation process used on the SIEM VM. As always, save your username and password to a password manager!

- On the *SSH Setup* screen, be sure to select *Install OpenSSH server*
- On the *Featured Server Snaps* screen, Select *Done* and hit enter to move on to the *Installation complete* phase
- Once the installation has finished, close the virtual console and select *Power off the machine*

10.7.2.5 Additional Virtual Machine Settings – IPS VM

Now that Ubuntu Server 20.04 is installed on the IPS VM, students will need to make a couple of minor changes to the IPS virtual machine before booting it to do network connection checks and perform updates. Just like with the SIEM VM, students will need to modify the Boot order under the *Motherboard* tab of the *System* menu, and remove both the IDE controller and Virtual CD/DVD drive from the *Storage* menu. For guidance on how to perform these actions, refer to [section 10.7.2.2](#) (pp. 263-264). The instructions are exactly the same for the IPS VM. Refer to *fig. 10-52* for further guidance.



10-52: Once Ubuntu Server 20.04 is installed, access the IPS virtual machine's settings menu through the *VirtualBox Manager* window. Under the *System* menu, on the *Motherboard* tab, set the *Boot Order* to where the *Hard Disk* is the first and only object the virtual machine is allowed to boot from. Then under the *Storage* menu, remove the Virtual optical disk and IDE controller. Once finished, click *OK* in the bottom right corner to apply these new settings.

10.7.2.6 Booting the IPS VM for the first time

Power the IPS VM back on, and once Ubuntu has finished starting up and performing its first-time boot routines, log in with the username and password assigned on the *Profile setup* screen during the install. Just like with the SIEM VM, students will run the following three commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output from the `ip -br a` command will differ slightly, because the IPS VM has more network interfaces than the SIEM VM, but aside from that, the output from `nslookup` and `curl` should be more or less identical to the output of these commands from the SIEM VM. See *fig. 10-53* for an example on what the output of these commands should look like.

```
Ubuntu 20.04 LTS ips tty1
ips login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@ips:~$ ip -br a
lo                UNKNOWN        127.0.0.1/8  ::1/128
enp0s3            UP             172.16.1.4/24 fe80::a00:27ff:feeb:2de9/64
enp0s8            DOWN
enp0s9            DOWN
ayy@ips:~$ nslookup www.google.com
Server:          127.0.0.53
Address:         127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.8.196
Name:   www.google.com
Address: 2607:f8b0:4009:815::2004
ayy@ips:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Thu, 23 Jul 2020 17:21:51 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Thu, 23 Jul 2020 17:21:51 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-23-17; expires=Sat, 22-Aug-2020 17:21:51 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=No2uEqnF70q9zD3pzs0rY1b9m4o1HDDzP4BzZ1ULDM2ia7uXqWv97cWdZNOfc2JxijI_BXyxhRfuF2EEvFV50ssKkaJRIZPxm4TbIdfzAihP6aW6FsTqHu6Kif6j75q06iuFFU-UP0oA73r0ytPyD314nvxBKnu1_rqEmla0Sic; expires=Fri, 22-Jan-2021 17:21:51 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

10-53: Just like with the SIEM VM, students will log in to the IPS virtual machine and run a couple of network diagnostic commands. The output from curl and nslookup commands should be more or less identical to the output on the SIEM VM, but the ip -br a command will produce a few more lines of content. Ignoring the lo (loopback) interface, there should be three interfaces. Only one of them should have the status of UP. That interface should be the interface assigned to the LAN/Management network segment, with the IP address 172.16.1.4

After running these commands to confirm the IPS VM has been assigned the proper IP address, can resolve hostnames, and has HTTPS connectivity, run the commands:

```
sudo su -
apt-get update
apt-get -y dist-upgrade
init 6
```

In order to become the root user, install updates on the IPS VM (and confirm the Squid proxy server is proxying the IPS VM's HTTP requests), then reboot after the system is done installing those updates.

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [306 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [114 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [7612 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [136 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [4792 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Fetched 889 kB in 1s (1269 kB/s)
Reading package lists... Done
root@ips:~# apt-get -y dist-upgrade_
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@ips:~# init 6

```

10-54: These commands are identical to the ones students ran on the SIEM virtual machine, and they serve the same purpose for the IPS VM: become the root user, check for updated packages, install those updates, then reboot the system.

Note: If you're having problems with your apt-get commands failing, refer back to the sidebar conversation on p. 268-269, *Help! My apt-get commands are failing!* For further guidance. Students can follow the exact same steps laid out for the SIEM VM to troubleshoot the problem.

10.7.2.7 Installing Kali Linux on the kali VM

Now that the SIEM and IPS virtual machines are out of the way, next up is the kali VM. Power on the kali VM, and if prompted by the *Select start-up disk* window, click *Cancel* to continue. A boot menu appears with a number of options. Using the arrow keys, highlight *Install* and hit enter.

Similar to the Ubuntu installer, the first screen, titled *Select a language*, asks users to choose the language they want to use for their installation. The default setting is *English*, use the arrow keys to highlight another language as necessary, then hit enter. The next screen, *Select your location*, asks users to choose what country, territory or area in which they are located. This screen defaults to *United States*. Use the arrow keys to change this value as necessary, and hit enter to continue. Next up is the *Configure the keyboard* screen, that asks the user what keymap to use for their installation. The default setting is *American English* and can be changed with the arrow keys. After highlighting a keymap, hit enter to proceed.

The installer begins loading other components of the installation process. Afterwards, it will attempt to get an IP address. The pfSense DHCP server should give it an IP address through the

OPT1 DHCP server, but students will not be able to confirm if the IP address 172.16.2.2 was correctly assigned until after the operating system is installed. The next screen, titled *Configure the network*, prompts users to enter a hostname for the system. Students should use the default hostname *kali*. Hit the enter key to continue to the next screen that prompts for a domain name. Again, students may hit enter and accept the default setting.

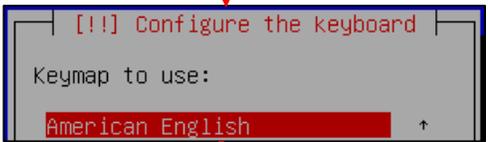
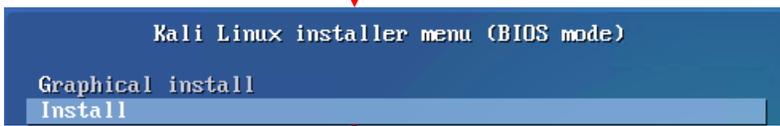
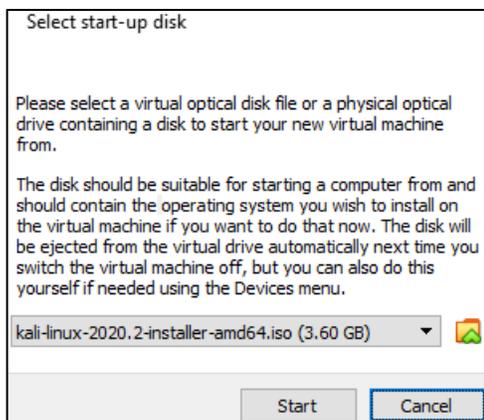
The *Set up users and passwords* screen appears. The first window asks for the full name of the user to be created. Type in the full name of the user account, and hit enter to continue to the next screen, that prompts for a username students will use to log in to the system. After typing in a username, hit enter to be prompted to create a password for this account. After hitting enter again, you'll be prompted to enter the same password again to confirm your choice. Enter the same password and hit enter to continue the installer. Just like with the SIEM and IPS virtual machines, be sure to save the username and password to your preferred password manager.

Next up is the *Configure the clock* dialogue. The installer will reach out to its preferred NTP servers to get the current time, then ask the user to select the time zone in which they are located. Use the arrow keys to choose your time zone, and hit enter to continue.

The *Partition disks* screen appears and asks users to select a partitioning method. Highlight the selection *Guided – use entire disk*, and hit enter. Users are then prompted to select the disk to partition. Since there is only a single virtual disk for the kali VM, hit enter to proceed. The next screen prompts students to select the partitioning scheme. Highlight the option *All files in one partition (recommended for new users)* and hit enter. Users are asked to confirm their choices on the next screen. Highlight the option *Finish partitioning and write changes to disk*, and hit enter. One final pop-up appears to annoy you, asking if students are sure they want to proceed, highlight *<Yes>* to confirm your choices, and press enter to continue.

After a moment or two, a window labeled *Software selection* appears. As the name implies, this screen allows users to pick additional software packages to install. Accept the default selections by pressing the tab key to highlight *<Continue>*, and hitting enter. The next portion of the installer retrieves and installs the requested packages. This portion of the installation may take some time, depending on internet speed and virtual machine performance.

After some time has passed, a new prompt appears labeled, *Install the GRUB boot loader on a hard disk*, asking if users want to install the GRUB boot loader. This is a necessary component in order to boot the virtual machine, so highlight *<Yes>*, and press the enter key to continue. The next screen asks what partition to install the boot loader to. Seeing as how there is only one partition available, highlight it, and hit enter to proceed. After a moment or two passes, students are prompted to remove the installation media, and reboot the virtual machine to complete the installation. Just like with the SIEM and IPS virtual machines, close the virtual console, and select the option labeled *Power off the machine*.



Continued to *fig. 10-56*

10-55: If prompted by the *Select start-up disk* dialogue, hit *Cancel* to continue booting the kali VM into the Kali Linux installer. The first screens have users select their preferred language, location, and keyboard keymap.

Continued from *fig. 10-55*

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

kali

<Go Back> <Continue>

Domain name:

localdomain

<Go Back> <Continue>

Continued to *fig. 10-57*

10-56: The next few screens configure some of the network settings. Students are prompted to enter a hostname and domain name. Students should use the default hostname of *kali*, and default domain name of *localdomain*.

Continued from *fig. 10-56*

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

ayy lmao

<Go Back> <Continue>

Username for your account:

ayy

<Go Back> <Continue>

Choose a password for the new user:

[] Show Password in Clear

<Go Back> <Continue>

Re-enter password to verify:

[] Show Password in Clear

<Go Back> <Continue>

Continued to *fig. 10-58*

Title:	kali VM
Username:	ayy
Password:	*****
URL:	172.16.2.2
<input type="checkbox"/> Expires:	7/24/2020 11:35 AM
<input checked="" type="checkbox"/> Notes:	credentials for the kali VM

10-57: Similar to the *Profile Setup* screen in the Ubuntu installer, the Kali Linux installer features a series of prompts to create a user account for the system. Be sure to save the credentials to your preferred password manager when finished.

Continued from *fig. 10-57*

[!] Configure the clock

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

Eastern

[!!] Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk

Select disk to partition:

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

Partitioning scheme:

All files in one partition (recommended for new users)

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

#1	primary	81.6 GB	f	ext4	/
#5	logical	4.3 GB	f	swap	swap

Undo changes to partitions
Finish partitioning and write changes to disk

Write the changes to disks?

<Yes> <No>

Continued to *fig. 10-59*

10-58: After setting the time zone, students will have to configure the partitioning scheme for the install. The highlighted options above should be selected by default. If not, use the arrows to select them, and press enter to continue.

Continued from *fig. 10-58*

```
[!] Software selection

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

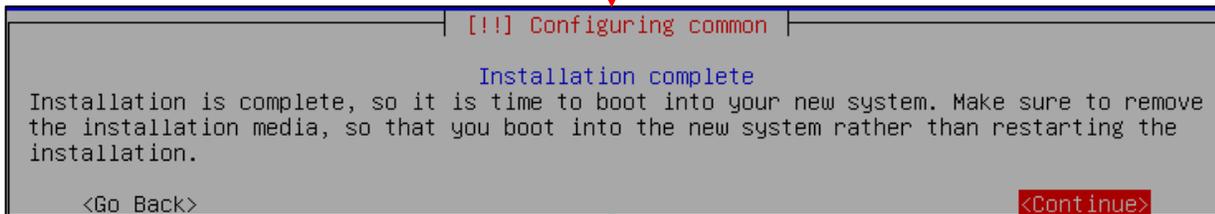
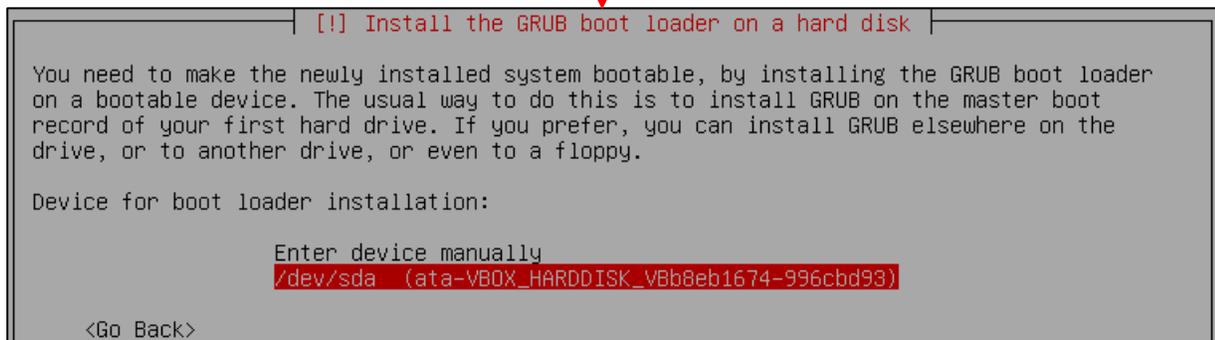
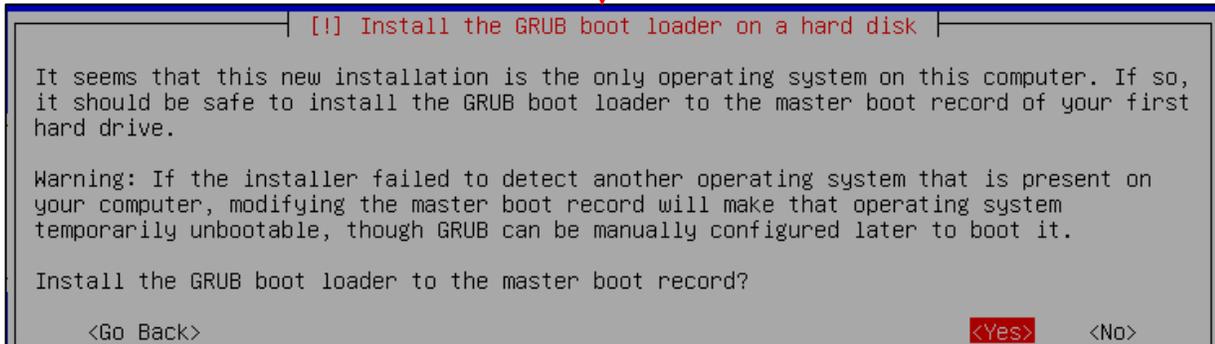
[*] Desktop environment [selecting this item has no effect]
[*] ... Xfce (Kali's default desktop environment)
[ ] ... GNOME
[ ] ... KDE Plasma
[*] Collection of tools [selecting this item has no effect]
[*] ... top10 -- the 10 most popular tools
[*] ... default -- recommended tools (available in the live system)
[ ] ... large -- default selection plus additional tools

<Continue>
```

Continued to *fig. 10-60*

10-59: On the *Software selection* screen, press the tab key to highlight *<Continue>* and hit enter to accept the default software package selections.

Continued from *fig. 10-59*



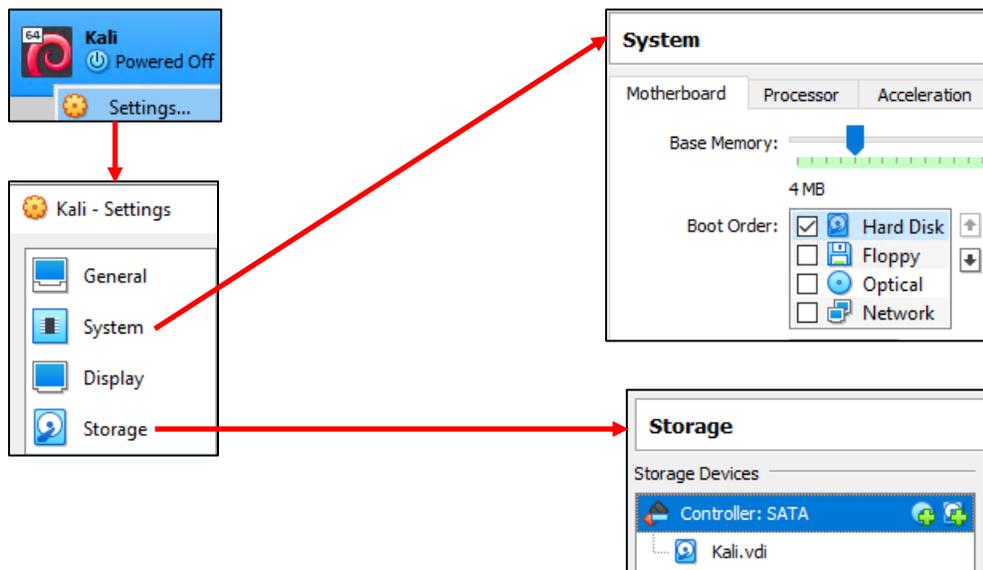
10-60: The final steps of the installation process. Install the GRUB boot loader to the only available disk on the system (this should be `/dev/sda`), wait for the screen labeled *Configuring common* to appear, then close the virtual console, choosing the option, *Power off the machine* to complete the install.

10.7.2.8 Additional Virtual Machine Settings – kali VM

By this point, the changes students need to make to the kali VM should be pretty routine. Right click the kali VM in the *VirtualBox Manager*, select *Settings...* and make the following changes:

- Edit the *Boot Order* list in the *System* menu, under the *Motherboard* tab. Ensure *Hard Disk* is the first item in the list, and the only item with a checkbox.
- In the *Storage* menu, remove both the virtual CD/DVD drive, and the IDE controller. *Kali.vdi* and *Controller:SATA* should be the only objects in the *Storage Devices* window

If necessary, guidance on how to perform these actions, can be found in [section 10.7.2.2](#) (pp. 263-264).



10-61: In the kali VM settings, Set the Boot Order to where on the Hard Disk is the first and only bootable media, then remove both the virtual optical drive, and the IDE controller.

10.7.2.9 Booting the kali VM for the first time

With those last-minute virtual machine settings applied, power the kali virtual machine back on. After a moment or two passes, students will be greeted with a graphical interface, asking for a username and password to log in. Enter the username and password supplied during the operating system install, and click *Log In* to continue.

On the top of the graphical user interface, there should be a menu bar with a few icons displayed. One of those icons is a small black window. Click on that icon to open a terminal session on the kali VM. With the terminal window open, run the same three commands that we ran on the SIEM and IPS virtual machines in order to confirm network connectivity is working as intended:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output of `ip -br a` should confirm that only a single interface (again, ignoring the `lo` interface) is installed on the system. If students copied the MAC address of adapter 1 correctly and created a static DHCP mapping on the OPT1 interface (through the pfSense webConfigurator), that interface should have the IP address 172.16.2.2. As with the SIEM and IPS virtual machines, if this is not the case, students should compare the MAC address of adapter 1 on the kali VM to the MAC address of the static DHCP mapping made on pfSense. **Make sure the mac addresses match, and that the mapping was created on the OPT1 interface.**

As with the SIEM and IPS VMs, `nslookup` confirms the ability of the kali VM to resolve hostnames through DNS, and the `curl` command verifies that the VM can make outbound internet connections over HTTPS. The output of these commands should be similar to the output displayed in *fig. 10-62* below.

While Kali Linux is slightly different from Ubuntu, we can still use *most* of the same commands utilized on the SIEM and IPS virtual machines to become root, check for updates, then reboot the system. **Run these commands in this exact order:**

```
sudo su -
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
apt-get update
apt-get -y dist-upgrade
init 6
```

Students may have noticed two new commands have been added here:

```
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
```

These commands are responsible configuring the apt package manager to use our HTTP proxy at 172.16.2.1:3128 on the *OPT1* interface of the pfSense VM. This is done by running the echo command, and redirecting its output (the > symbol) to the file /etc/apt/apt.conf.d/99local (a configuration file that the package manager will read when we run apt-get later). The second command, cat /etc/apt/apt.conf.d/99local, reads the contents of the file. If the output from the cat command reads:

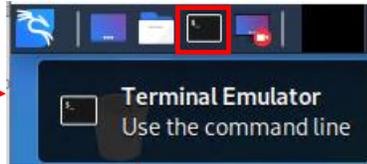
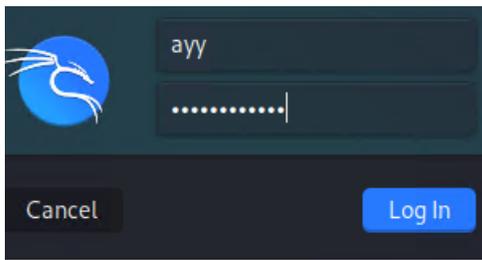
```
Acquire::http::Proxy "http://172.16.2.1:3128";
```

Then that means apt was successfully configured to use the HTTP proxy. If the output from the cat command displays anything else, then students should re-enter the echo command.

Note: If most of these commands look familiar, it's because they're very similar to the troubleshooting commands I recommended in the sidebar discussion [*Help! My apt-get commands are failing!*](#) (pp. 268-269) for the SIEM and IPS virtual machines. There are a few key differences with the kali VM to be aware of, but for the most part, the troubleshooting steps laid out are the same as the steps I laid out in this section. Here are the key differences to be aware of:

- Make absolutely sure you are redirecting the output of the echo command to the file /etc/apt/apt.conf.d/99local. ***It must be that exact file, in that exact location.***
- The kali VM doesn't need the second line, Acquire::https::Proxy "http://172.16.2.1:3128";
- Make absolutely sure to specify http://172.16.2.1:3128 as the proxy address for the kali VM.

After running these commands to configure the package manager, students should be able to run the remaining commands just like on the SIEM and IPS virtual machines. Bear in mind that Kali Linux is subject to frequent updates, and that some of those updates can be quite large. This means that depending on the performance of the Kali VM, and internet connection speeds, downloading and installing updates may take some time to complete.



```
ayy@kali:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
eth0              UP                172.16.2.2/24 fe80::a9d9:817a:46ee:84c3/64

ayy@kali:~$ nslookup www.google.com
Server:          172.16.2.1
Address:         172.16.2.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:804::2004

ayy@kali:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 24 Jul 2020 19:55:47 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 24 Jul 2020 19:55:47 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-24-19; expires=Sun, 23-Aug-2020 19:55:47 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=rC1q-094PKdmAIZC2ajgCkpdGrGdulzdaxnJR2Cui-HYKBgbjzh_qCz6G5tJYoetE_Uc7rscR53x4Gri7HE3k_gu9h2BKh6etyF0hGD0iat3FF22oe-4VngjLFAdGEY3XTecVHB8iJ5Qw2qUVjmmN7oZGeRUSj1mXJ8ulaLkRY; expires=Sat, 23-Jan-2021 19:55:47 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@kali:~$ sudo su -

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for avv:
root@kali:~# echo 'Acquire::http::proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
root@kali:~# cat /etc/apt/apt.conf.d/99local
Acquire::http::proxy "http://172.16.2.1:3128";
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@kali:~# init 6
```

10-62: Login to the kali VM, configure the apt package manager to use the SQUID HTTP proxy on *OPT1* of the pfSense VM. Afterwards, install the latest operating system updates, then reboot the virtual machine.

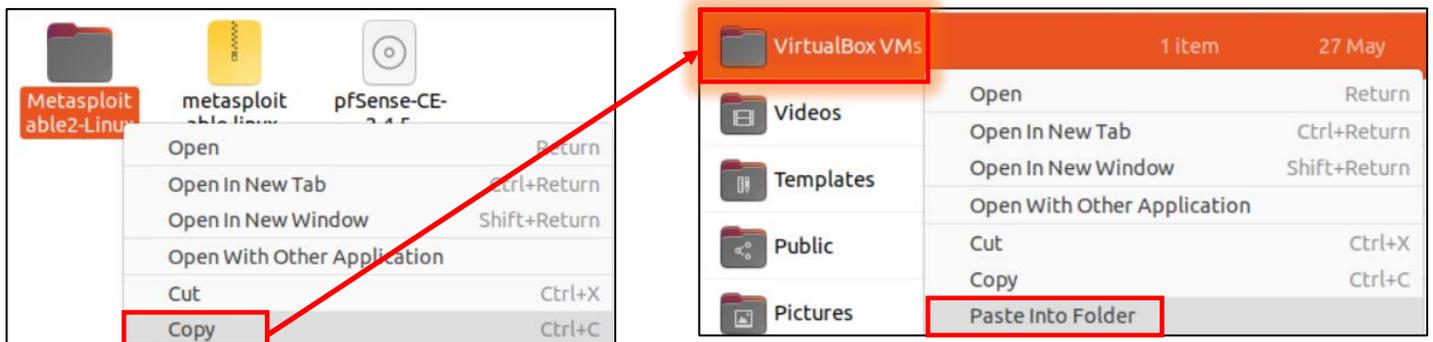
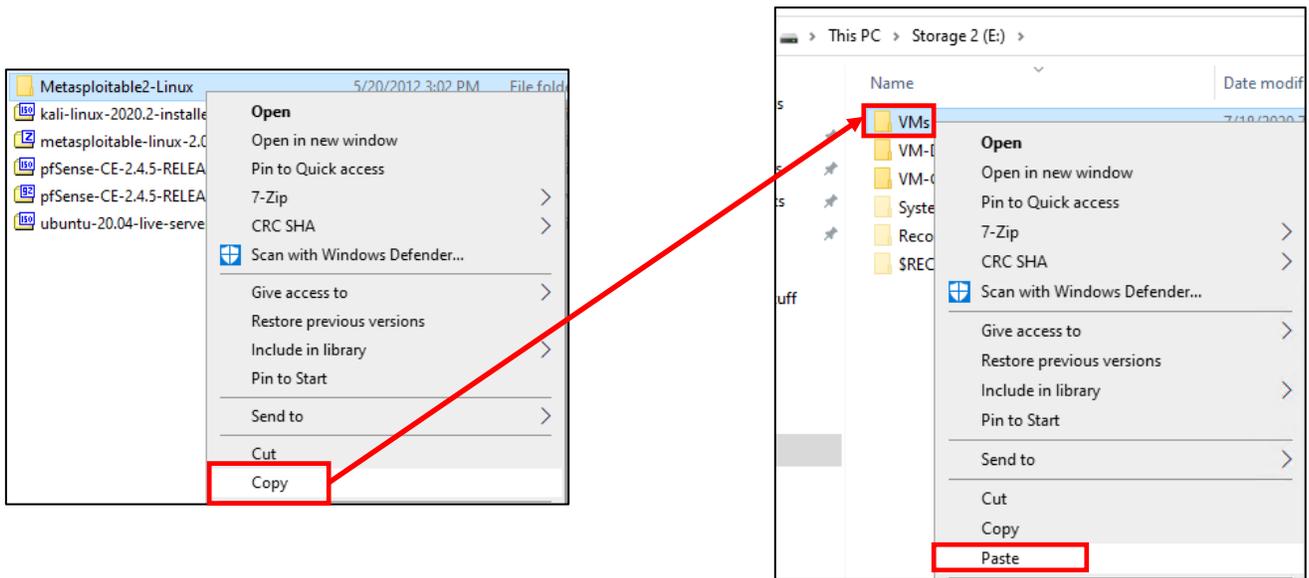
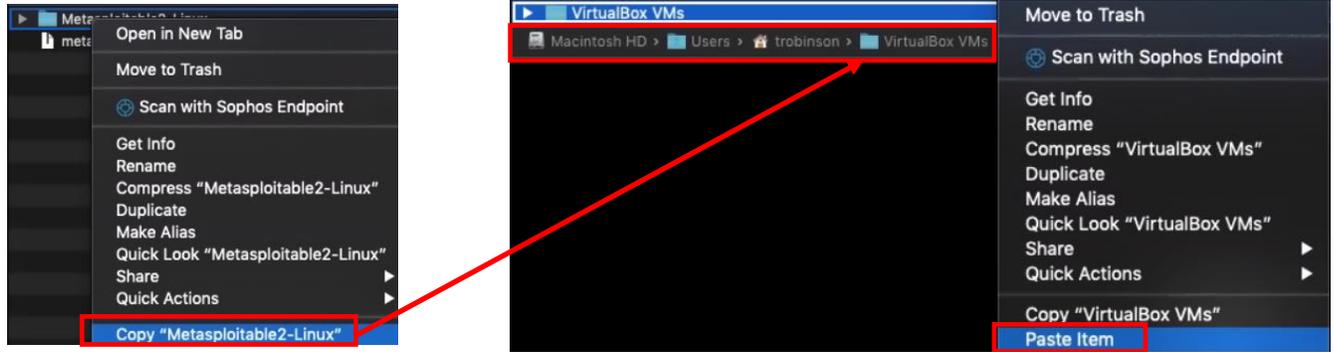
10.7.3 Metasploitable 2

The Metasploitable 2 VM is slightly different than the other VMs. It's pre-made with an operating system already installed, so all that needs to be done is to move its files to the *Default Machine Folder*, and import it into VirtualBox, so that it is listed under the *VirtualBox Manager*. Once there, we can edit the virtual machine settings, record the MAC address to make a static DHCP mapping, then boot the virtual machine to confirm everything works properly.

Before we can proceed, students must perform the following tasks:

- In section 10.4 (pp. 208-209), students were instructed on how to access the VirtualBox *Preferences* menu. Under *General*, review the *Default Machine Folder* setting. Students will need to copy the decompressed Metasploitable 2 files to this directory.
- Way back in chapter 1, section 1.8 (pp. 33-35), students were instructed on how to use compression utilities on Windows, Linux or MacOS to decompress files. Students should have already decompressed the Metasploitable 2 VM, `metasploitable-linux-2.0.0.zip`. This file, when decompressed, should yield the directory `Metasploitable2-Linux`. This directory needs to be moved or copied to the *Default Machine Folder*.

Most operating systems with a graphical user interface (e.g., Windows, MacOS, and Linux) feature some sort of a file browser that allow users to copy and paste files/folders by dragging and dropping from one folder to another, or by right-clicking a file/folder, selecting *copy*, then navigating to the destination folder, right-clicking and selecting *paste*. In other cases, command line commands can be used to perform the same task (e.g., the Linux/MacOS `cp -r` and/or `mv` commands). Students may use whatever means they are most comfortable with to place the `Metasploitable2-Linux` directory in their designated default machine folder. Figure 10-63 below demonstrates how to copy and paste the `Metasploitable2-Linux` directory on MacOS, Windows, and Linux operating systems.



10-63: This illustration shows how MacOS, Windows and (most) Linux distributions with a graphical interface have some sort of file browser that supports copy and paste, and operates on the same principle: right click the directory, paste into the destination. For MacOS, the *Default Machine Folder* is usually `/Users/[username]/VirtualBox VMs`. For Linux users, it's usually `/home/[username]/Virtualbox VMs`. Finally, for Windows users, the default directory is usually set to `C:\Users\[username]\Virtualbox VMs`, but in the illustration above, I changed it to `E:\VMs`.

10.7.3.1 Importing Metasploitable 2

Open the *VirtualBox Manager* window, and begin the *Create Virtual Machine* wizard. (Recall that this can be done by clicking *Tools* above the virtual machine inventory list, then clicking the large blue spiked circle labeled *New*, or by clicking *Machine* in the navigation menu, then clicking *New...*) Use the following settings on the *Name and operating system*, and *Memory size* screens:

Name	Metasploitable 2
Machine Folder	Default location
Type	Linux
Version	Ubuntu (64-bit)
Memory	512MB

On the *Hard disk* screen, students will instruct VirtualBox to use Metasploitable 2's VMDK file as it's virtual hard disk. Select the radio button labeled *Use an existing virtual hard disk file*, then click the small folder icon with the green "^" to open the *Hard Disk Selector* window. Click the *Add* button to open your host operating system's file browser.

Navigate to the *Default Machine Folder*, where the *Metasploitable2-Linux* directory should be located. Double click on the *Metasploitable.vmdk* file to select it. For example, on my Windows system, the complete file path was *E:\VMs\Metasploitable2-Linux\Metasploitable.vmdk*. This will cause the file to appear in the *Hard Disk Selector* window, under a list labeled *Not Attached*. Make sure *Metasploitable.vmdk* is highlighted (it should be, by default), then click *Choose*. This will bring students back to the *Hard disk* screen, with *Metasploitable.vmdk* chosen as the existing virtual hard disk to use. Click the *Create* button to continue.

If all the actions were performed correctly, Metasploitable 2 should be listed under the virtual machine inventory list in the *VirtualBox Manager* window.

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name:

Machine Folder:

Type:

Version:

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024 MB**.

MB

4 MB 32768 MB

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

Do not add a virtual hard disk

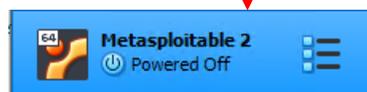
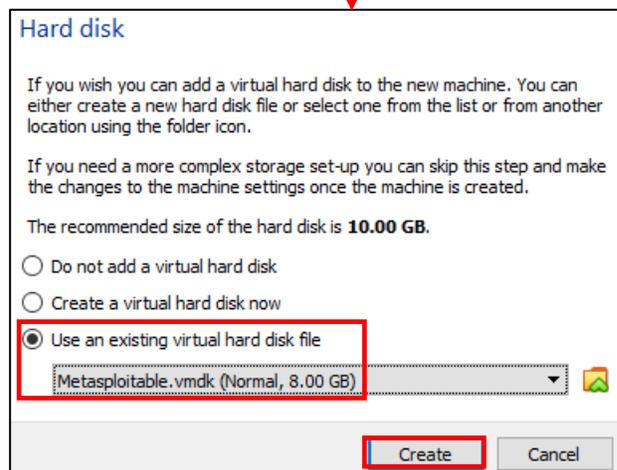
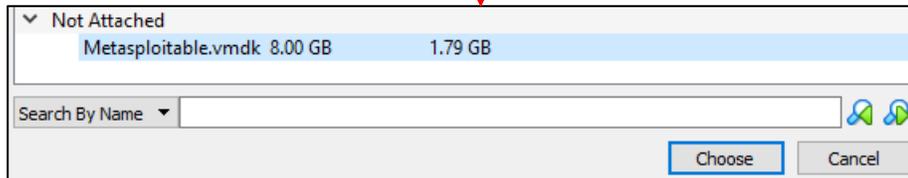
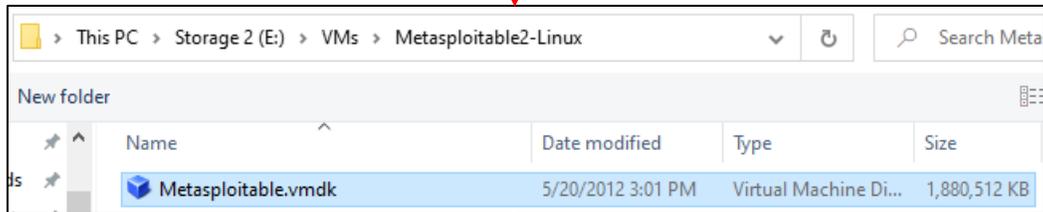
Create a virtual hard disk now

Use an existing virtual hard disk file

Continued to *fig. 10-65*

10-64: Create the Metasploitable 2 VM at first as though it were a normal virtual machine. On the *Hard disk* screen, Select the option to use an existing virtual hard disk, then click the little folder icon with a green carat symbol (^) over it.

Continued from *fig. 10-64*



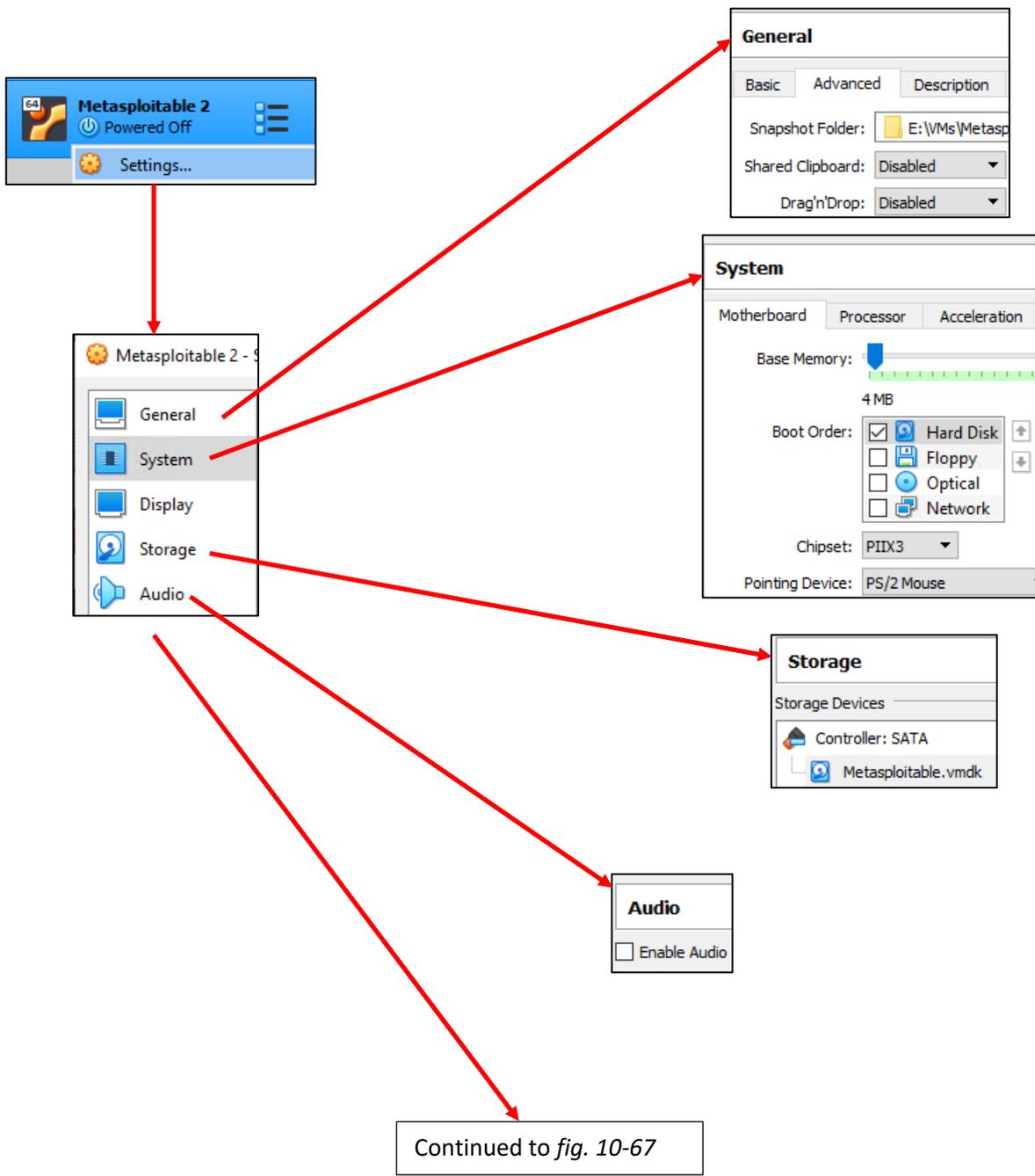
10-65: On the *Hard Disk Selector* window, Click the *Add* button, and browser to the *Metasploitable2-Linear* directory. Select the *Metasploitable.vmdk* file, and it appears in the *Hard Disk Selector* window. Highlight *Metasploitable.vmdk* under the listing *Not Attached*, click *Choose*, then back on the *Hard disk* screen, click *Create* to add the *Metasploitable 2* VM to the virtual machine inventory.

10.7.3.2 Adjusting Metasploitable 2 VM settings

Before powering on the Metasploitable 2 VM, students will need to edit the virtual machine settings. The main difference between Metasploitable 2, and the other virtual machines is that students can perform the pre and post operating system install adjustments all at once, since Metasploitable 2 is a pre-built VM, with the operating system already installed. By this time, students should be comfortable with editing the settings of their virtual machines. As always here is a list of configuration settings to change:

- Open the *VirtualBox Manager* window, right click on Metasploitable 2 and click *Settings*. Once there, adjust the following configuration settings:
- On the *General* sub-menu, under the *Advanced* tab, ensure that the *Shared Clipboard*, and *Drag'n'Drop* drop-downs are set to *Disabled*.
- Under *System* on the *Motherboard* tab, modify the boot order. Disable all bootable devices except the *Hard Disk* option, then adjust *Hard Disk* to be the first option at the top of the list. Change the *Pointing Device* drop-down option to *PS/2 Mouse*.
- For *Storage* settings, remove both the virtual CD/DVD drive, and IDE controller. Students can refer to section 10.7.2.3 for more detail instructions if a refresher is needed. The only entries under the *Storage Devices* window should be the SATA controller, and the *Metasploitable.vmdk* virtual disk.
- Under *Audio*, uncheck the *Enable Audio* checkbox
- On the *Network* settings screen, ensure that *Adapter 1* is the only tab with the *Enable Network Adapter* checkbox checked. Set the *Attached to* drop-down to *Internal Network*, and in the *Name* input box, enter "intnet1". Copy the contents of the *MAC Address* field and use them to create a Static DHCP mapping under *Services > DHCP Server* on the pfSense Webconfigurator, on the *OPT1* interface. Assign Metasploitable 2 the IP address 172.16.2.3. Finally, ensure that the *Cable Connected* checkbox is checked.
- Under *Serial Ports*, ensure that *Enable Serial Port* is unchecked on ports 1 through 4.
- For *USB* settings, uncheck the *Enable USB Controller* checkbox
- Finally, under *Shared Folders*, ensure there are no shared folders configured.

After students are done editing all of these settings, they may click *OK* in the bottom right of the *Metasploitable 2 – Settings* window, in any one of the sub-menus to apply all of their modifications at once.



10-66: Before booting Metasploitable 2, open its *Settings* menu to make some necessary changes. At this point, making and checking these configuration settings should be second nature, but if students need a refresher, take a look at [section 10.7.1](#) (pp. 249-254), and [10.7.2.2](#) (pp. 263-264).

Continued from *fig. 10-66*

Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

Enable Network Adapter

Attached to: Internal Network

Name: intnet1

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 08002731C52B

Cable Connected

Services / DHCP Server / OPT1

Mappings for this Interface

MAC address	IP address	Hostname	Description
08:00:27:69:e9:0c	172.16.2.2	kali	DHCP mapping for kali VM
08:00:27:31:c5:2b	172.16.2.3	metasploitable2	DHCP mapping for metasploitable 2

Serial Ports

Port 1 Port 2 Port 3 Port 4

Enable Serial Port

USB

Enable USB Controller

Shared Folders

Shared Folders

Name	Path	Access	Auto Mount	At
Machine Folders				

10-67: The other half the Metasploitable 2 VM settings. Pay very close attention to the *Adapter 1* settings the *Network* sub-menu. The *Attached to* drop-down should be set to *Internal Network*, and the *Name* to *intnet1*. Additionally, make sure to record the contents of the *MAC Address* field, and use that to make a static DHCP mapping under the *OPT1* interface on the pfSense webConfigurator. Metasploitable 2 should be assigned the IP address 172.16.2.3. If students need a refresher, refer to chapter 14, [section 14.3.4.1](#) (pp. 690-692) for instructions on how to set up a static DHCP mapping.

Why aren't we doing connectivity checks?

Some of you may be wondering why we aren't doing connection checks or any of the stuff we did we for the SIEM, IPS, or Kali VMs, like checking the IP address or attempting to connect outbound. Well, that's because right now, the metasploitable 2 VM doesn't have an IP address. Don't worry, its intentional, and you'll be fixing this later. The reason metasploitable 2 doesn't have an IP address is that it's connected to the *IPS 2* network. While technically the IPS 2 network shares the same network subnet as IPS 1, and logically it's all a part of the OPT 1 network, IPS 2 is its own physical network segment, and entirely separate from the IPS 1 network. Without something to physical connect the IPS 1 and IPS 2 networks together, the IPS 2 network is entirely isolated.

Remember the network diagram back in [chapter 6](#) (p. 58)? The *IPS 2* network relies on the IPS virtual machine being fully configured and running either Snort or Suricata in AFPACKET bridging mode. No network bridge, no network connectivity. That means no IP address from the DHCP server, either. Run the command `ifconfig -a`. Sure, there's an IPv6 address configured but that's an internal/link-local address. You'll be fixing this problem later when you install either Snort or Suricata to the IPS virtual machine in chapter 17.

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:31:c5:2b
          inet6 addr: fe80::a00:27ff:fe31:c52b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4230 (4.1 KB)
          Base address:0xd010 Memory:f1200000-f1220000
```

10-69: the output from `ifconfig -a` (without the `lo` interface output). `eth0` never got assigned an IPv4 address from DHCP because there is no physical connectivity between the IPS 1 and IPS 2 segments. We'll be solving this problem later when students install Snort or Suricata to the IPS virtual machine.

10.8 Snapshots

The next (and final) task for students to perform will be creating baseline virtual machine snapshots for the entire lab environment. Snapshots instruct the virtual machine's hypervisor to gather information about the VM's current state, and save it. Later on, if there is a problem with the virtual machine such as a malware infection, or a configuration problem that cannot be diagnosed, users can choose to restore the virtual machine to its state in the past, when the snapshot was initially taken.

Virtual machines can have more than one snapshot, with the only limit being disk space required to hold the snapshots. It's extremely important to note that **virtual machine snapshots are not a substitute for backups**. If students plan on running virtual machines with important data that they cannot afford to lose, snapshots are not a substitute for backing up important files and data.

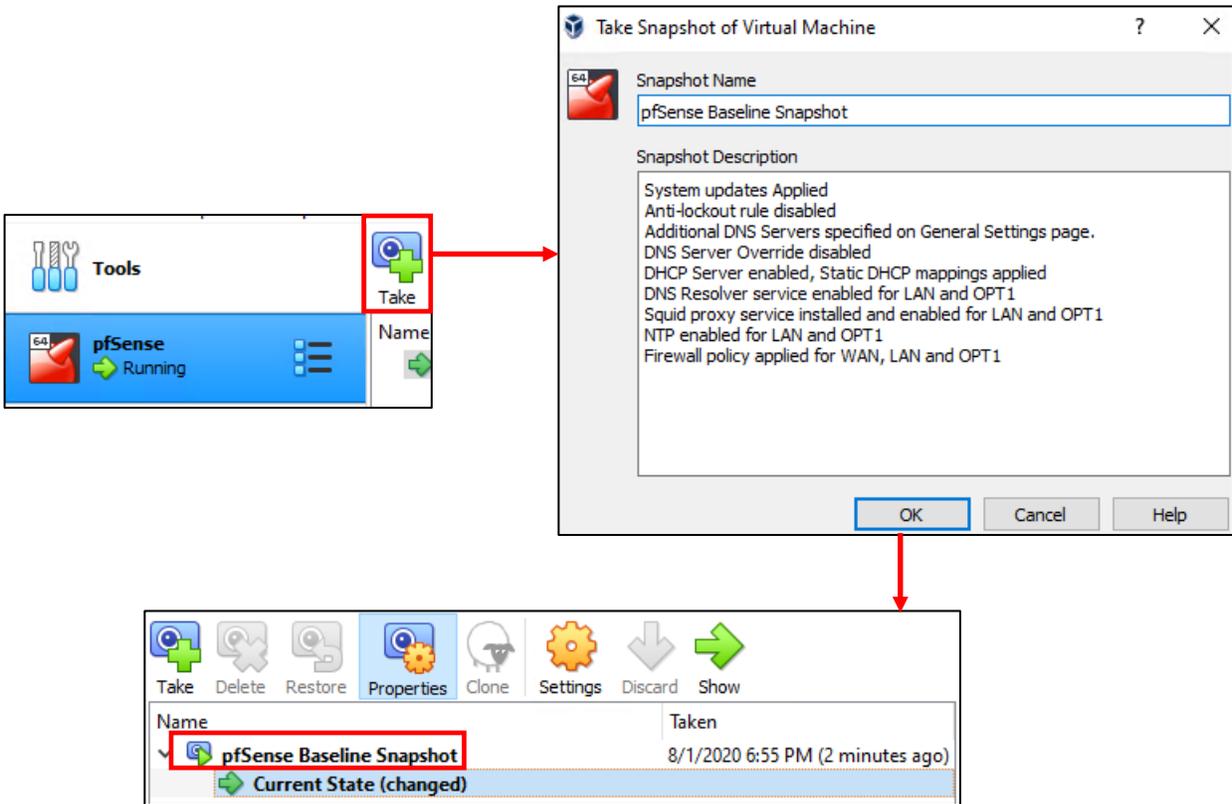
In this section, students will walk through the process of creating a baseline snapshot for the pfSense VM. Afterwards, it will be left as an exercise to the students to repeat the process for the SIEM, IPS, kali, and Metasploitable 2 virtual machines. Once finished, students will be ready to move on with the configuration of their lab environment.

10.8.1 How to Take a VM Snapshot

Open the *VirtualBox Manager* window, and click on the virtual machine to snapshot in order to highlight it. Along the top-right of the window, to the left of the big green start button used to power on the virtual machine, are a series of buttons labeled *Take*, *Delete*, *Restore*, *Properties*, and *Clone*. Click the button labeled *Take*, and a pop-up window labeled *Take Snapshot of Virtual Machine* appears, with two input boxes labeled *Snapshot Name*, and *Snapshot Description*. For the Snapshot Name, I recommend entering something descriptive, such as pfSense Baseline Snapshot. As for the Snapshot Description, I recommend documenting the current state and configuration of the virtual machine. For example, here is the description I used:

```
System updates Applied
Anti-lockout rule disabled
Additional DNS Servers specified on General Settings page.
DNS Server Override disabled
DHCP Server enabled, Static DHCP mappings applied
DNS Resolver service enabled for LAN and OPT1
Squid proxy service installed and enabled for LAN and OPT1
NTP enabled for LAN and OPT1
Firewall policy applied for WAN, LAN and OPT1
```

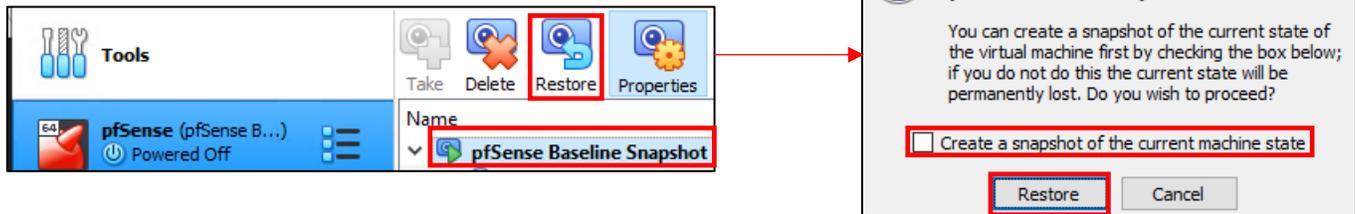
After entering a name and description for your snapshot, click the *OK* button. Underneath the snapshot operation buttons, there is a window with the fields *Name*, and *Taken*. The new snapshot students just took should appear, along with the default state of the virtual machine, named Current State. Congrats, the pfSense VM now has its first snapshot.



10-70: Highlight the target VM, click the *Take* button, enter a name and description for the snapshot, and click *OK*. Depending on the speed of your disk drive, this may take a moment or two.

10.8.2 Restoring a Snapshot

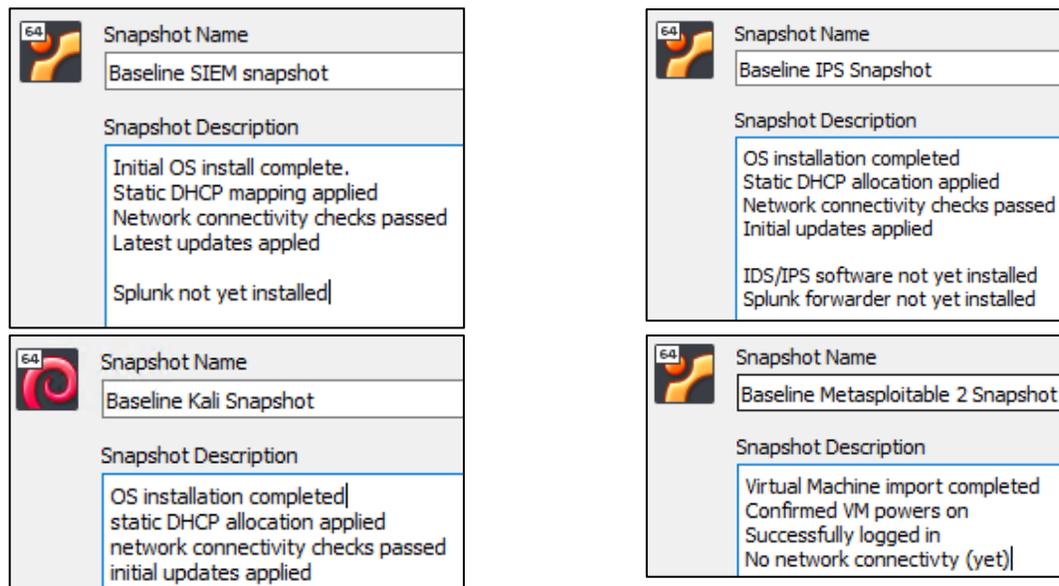
In order to restore a snapshot, the target VM must be powered off. So once again, using the pfSense VM as an example, power the virtual machine off. The fastest way to do this would be to close the pfSense VM's console, and select the *power off the machine* option. With the VM powered off, open the *VirtualBox Manager* window, click on the pfSense VM in the virtual machine inventory to highlight, and locate the pfSense Snapshot taken earlier, under the snapshot operation buttons. Left click on the name of the snapshot to highlight it, then click the restore button. A confirmation pop-up window appears. Uncheck the checkbox labeled *Create a snapshot of the current machine state*, then click *Restore* to begin the snapshot restoration process. After its completed, power the virtual machine back on.



10-71: The Snapshot restore process is even simpler than the creation process. The only thing to remember is that the VM must be powered off before attempting to restore a snapshot.

10.8.3 Snapshot the SIEM, IPS, Kali and Metasploitable 2 virtual machines.

Now that students understand how to create virtual machine snapshots, it is highly recommended that they create baseline snapshots for the remaining virtual machines in their lab environment – the SIEM, IPS, kali, and metasploitable 2 virtual machines. In the chapters to come, there will be a lot of complicated configuration tasks that students will need to perform in order to enable different functionality for their environment. Having a baseline snapshot to fall back to in case there are problems completing a task is handy for troubleshooting purposes.



10-72: Now that students know how to create VM snapshots, apply that knowledge and make baseline snapshots for the other lab virtual machines. Having a baseline to fall back to in case something fails in the later chapters of this book is very important and will save students a lot of headaches.

10.9 Chapter Review

Students should have all 5 virtual machines created for the baseline lab environment, as well as baseline snapshots for all 5 virtual machines. It was a long journey to get to this point, but it's far from over. Here is a checklist of tasks to complete:

- Complete [chapter 15, *Routing and Remote Access for Hosted Hypervisors*](#), starting on p. 727. In this chapter, students will learn how to enable SSH access to their lab virtual machines from Windows, Linux or MacOS. This functionality is vital for finishing the IPS and Splunk setup guides more easily than through the VM console alone.
- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in [chapter 17, *Network Intrusion Detection*](#), starting on p. 980.

Note: Make sure that the virtual network adapters attached to *intnet* and *intnet1* on the IPS VM have promiscuous mode set to *Allow All*. **This is absolutely vital to ensuring the IDS/IPS software functions correctly.** Refer back to [section 10.7.1, pp. 249-254](#).

- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in [chapter 18, *Setting up Splunk*](#), starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out [chapter 19, *End of the Beginning*](#), starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out [chapter 20, *Extra Credit*](#), starting on p. 1055.

Chapter 11 – Disclaimer for "M1" macs and macOS "Big Sur"

In November of 2020, Apple made it official that their hardware lineup would be no longer be powered by Intel processors, but instead would switching to ARM-based processors that they will be fabricating in-house. Apple is referring to these new processors that will be powering their desktop and laptop computers as the "M" series, starting with the hot-off-the-presses M1 processor.

If you are planning on attempting this chapter using the new Apple hardware with the M series processors, you will not be able to proceed at this time. Without getting too deep into the weeds, the M series processors, as I mentioned above use the ARM CPU architecture. VMware Fusion is currently only supported on Intel CPU based mac hardware (but VMware is going forward with creating a version of VMware Fusion that *will* run on the new mac hardware – with a bunch of caveats²). Intel CPUs use the x86-64 architecture. Even if and when VMware creates a version of VMware Fusion Pro that runs on the newer M series mac hardware, the operating systems for our virtual machines that are used for the lab environment are all predominantly x86 operating systems. That means that, unless the maintainers create a new operating system release that specifically runs on the ARM CPU architecture, the virtual machines themselves will not be compatible with the new mac hardware.

Let's assume that you are using an Intel-based mac, instead. Well now, you have a whole other set of problems to navigate. MacOS 11.x, ("Big Sur" and later) have made significant changes to its virtual networking API, called the vmnet API³. Additionally, there have been significant changes to how it handles drivers and/or kernel extensions. In my experience, many of the network virtualization functions unique to VMware Fusion pro, that are required to create our lab environment (e.g., custom virtual networks and/or the ability to create or remove virtual network interfaces for these custom virtual networks) are in an extremely broken state as of July 25, 2021. VMware Fusion developers are extremely aware of these problems, and major patches are being planned to work around the mess Apple made regarding their virtualization and virtual network API. Until this patch is released, I recommend utilizing Oracle VirtualBox.

Students running macOS Catalina, trying to wait for the dust to settle around Big Sur are mostly safe for now, and can proceed through this chapter with no problems, but with a huge caveat: Apple (and most of the developers in their walled garden) are quick to remove support for "legacy" versions of macOS. On the average, Apple will only support a given version of macOS for three years from its release. This means that macOS Catalina's days are very numbered.

² <https://blogs.vmware.com/teamfusion/2021/04/fusion-on-apple-silicon-progress-update.html>

³ <https://developer.apple.com/documentation/vmnet>

This is a lot to read, can you please summarize it in plain English?

-If students have an Intel-based mac running macOS Catalina and VMware Fusion Pro 12.x, they may proceed through this chapter, and should not have any major problems, aside from the fact that the end-of-life support date is rapidly approaching...

-If students have an Intel-based mac running macOS Big Sur or newer, VMware Fusion Pro 12.x's virtual network functionality is extremely broken right now, due to Apple making sweeping changes to their virtual networking APIs, as well as how they handle drivers and/or kernel extensions. Watch for software updates to VMware Fusion Pro, promising to resolve these problems. In the meantime, Oracle VirtualBox is an alternative that works.

-If students are using a new mac with one of the ARM-based "M" series processors, it's not likely they will be able to proceed through this chapter. VMware is planning on writing a version of VMware Fusion for "M" series CPUs, but then the problem comes down to the operating systems of our virtual machines not being compatible with neither the hypervisor nor the hardware. Many Linux distro maintainers are working on releases of their distributions that are ARM compatible, but there are also many who are not.

Chapter 11 Patch Notes

- There were many times over the past year and a half that I was planning on scrapping this chapter in its entirety. I originally wrote this chapter on macOS Catalina (10.15.x) on Intel hardware. Little did I know of the absolute mess Apple was going to drop on everyone's lap, decided to totally upheave their driver model and virtual network API in macOS Big Sur (11.1.x). On top of that, Apple then decided that they were too cool for x86, and are transitioning to their own custom ARM processors. Hate is a strong word, and make no mistake, I hate Apple. If I ever write a new edition, I will never have anything to do with Apple hardware or software ever again.

- How long this chapter remains relevant and actually allows students to build a functional lab environment on MacOS depends on how long Apple continues to support their x86 hardware, as well as when VMware patches VMware Fusion to resolve the virtual networking problems.

-Apple's new permissions model introduced in Catalina (10.15) is positively horrendous. Picture the worst aspects of Microsoft's User Account Control (UAC). Now, imagine this company having made a series of commercials in the early 2000s making fun of Windows Vista, while somehow making their desktop operating system experience in 2020 demonstrably worse than a desktop operating system from the early 2000s. Now, picture UAC pop-ups for every. Single. Action. You perform with the software you want to use, as you use those functions. Sometimes you're lucky, and the application registers *some* of the permissions it wants under *System Preferences > Security & Privacy*, and never informs you. Other times, you get a surprise pop-up telling you as a user that you have to manually allow this permission or the requested

functionality will not work. I documented these changes and made students aware that this is a thing they'll have to contend with in macOS Catalina onward.

- In macOS Catalina, I encountered a weird bug attempting to toggle VMware Fusion's network editor option, *Require authentication to enter Promiscuous Mode*. Namely, the first time I attempted to uncheck this checkbox and disable it, either VMware Fusion, or the operating system fought me, and filling the checkbox back in after attempting to apply my settings. I'm not sure if it was because I added additional virtual networks and/or edited other virtual network settings, and I was attempting to apply all of those settings at once, but I had to resort to quitting out of VMware Fusion entirely, reopening the network editor, unchecking the *Require authentication to enter Promiscuous Mode* checkbox, then applying my settings again for the change to finally apply.

-Most top-of-the-line MacBooks start out with 256GB of SSD for internal storage. Bear in mind that macOS can easily consume 40-60GB of that space default, with few to no additional apps or data stored on the drive. Also factor in the loss of disk space due to formatting and SSD maintenance, and that doesn't leave much space at all. This wouldn't be so bad if Apple's elegant hardware design didn't involve soldering everything to the board, and including as few I/O ports for external storage as possible. This realization forced me to note that, depending on the storage configuration students have on their Apple hardware, it might actually be a bad idea to attempt to pre-allocate all of the space for your virtual machine disk files (VMDKs) in advance. In fact, students may need to consider reducing the maximum disk size for the SIEM, IPS and Kali VMs. I recommend this method in chapter 6, but thought it could bear repeating. Another alternative is to acquire an external SSD or hard drive and store the virtual machine files there, but student VMs may not perform as well using this method.

-As with the other chapters, a template file for doing asset management of their lab environment is provided to students.

-Detailed instruction is provided to users for copying the decompressed Metasploitable2-Linux directory to the default virtual machine directory to easily import the virtual machine, and ensure that the VM gets ran from the correct location, etc.

-To that effect, students are guided through the process of upgrading the virtual hardware of the metasploitable 2 VM.

Chapter 11: VMware Fusion Pro

VMware Fusion Pro is a hosted hypervisor available for MacOS hosts exclusively. Like the other VMware hypervisors, it's been around for a very long time. This chapter will cover Installing and configuring Fusion Pro.

11.1 Installation

In the subsections that follow, students will learn how to install Fusion Pro on MacOS. Visit the following link:

<https://www.vmware.com/products/fusion/fusion-evaluation.html>

And download the installer.



11-1: As they say, the journey of a thousand miles begins with a single step. Our first step is downloading the VMware Fusion Pro package for MacOS. As the download link implies, students will need to be running MacOS 10.15 or higher to install Fusion Pro.

Getting the most Virtualization for your Money

Bear in mind that VMware Fusion Pro is not a free product. The link I provided above is for a 30-day free trial. After that, VMware will demand that you pay for a license for the pro features, or you will automatically be downgraded to VMware Fusion Player. We'll talk about VMware Fusion Player in a moment.

Sometimes, if you're lucky, employers will either provide a discount or reimburse you for the cost of a Fusion Pro license. So, what should the rest of us do, who have to pay out of pocket? The best recommendation I can offer is to pay attention during the holidays, specifically "Black Friday" and "Cyber Monday". VMware will occasionally sell licenses for their products at a discounted rate. Additionally, they also provide a steep discount to users who sign up to vmware.com with a ".edu" e-mail address. Through the "VMware Academic Program".

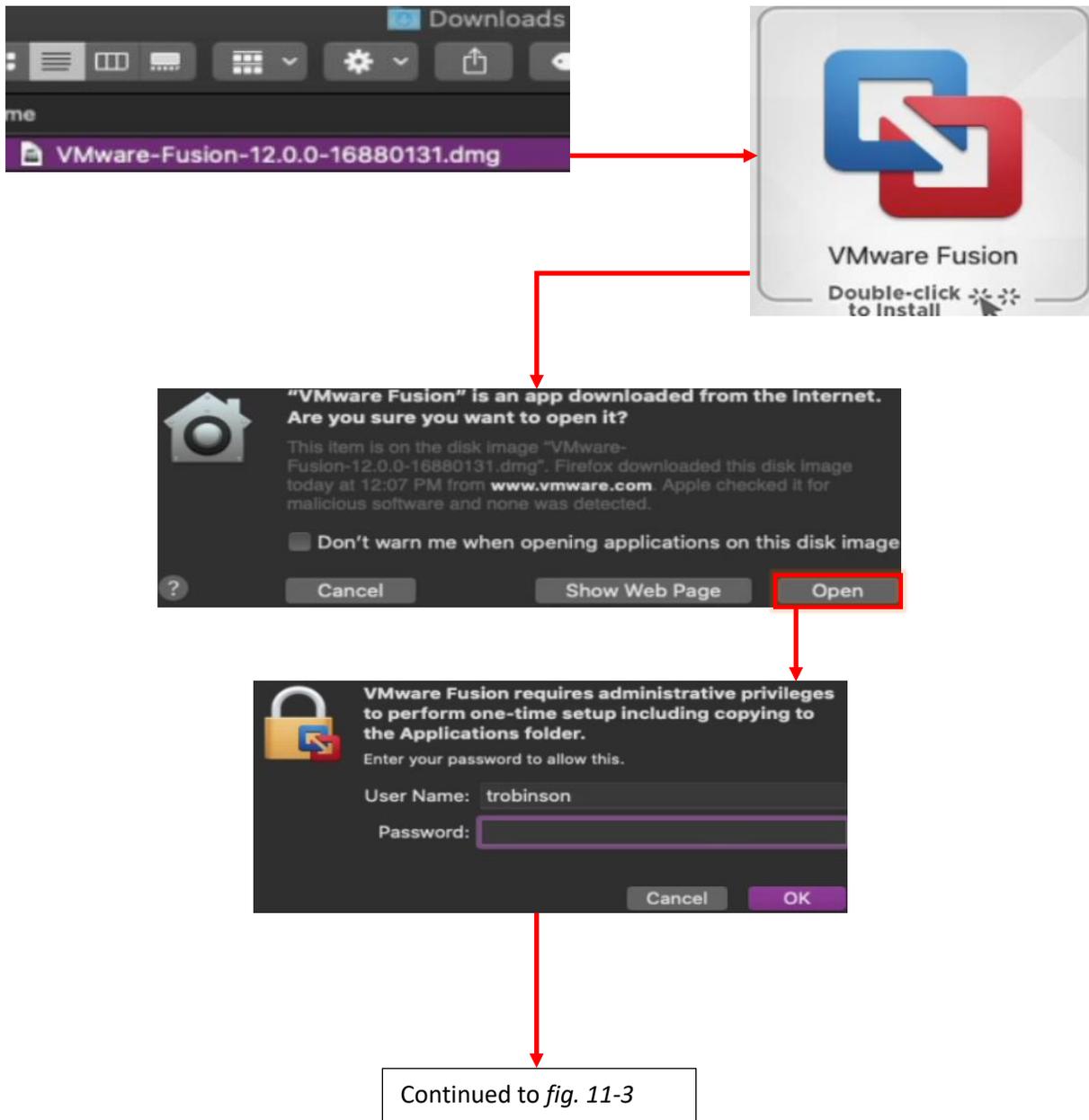
Once you've downloaded the Installer .dmg file, double click on it. This will mount the dmg file, causing a window to pop up with a large button with the text *VMware Fusion Double-click to Install*. Double click the button to begin the installation process. A pop-up will appear asking students if they're sure they want to open the application. Click the *Open* button to continue. Not content with one pop-up to impede progress, another pop-up will appear asking students to enter the password for their user account to proceed. Enter your password and click OK to finally proceed to the installer.

As always, the first screen that appears is a license agreement that students are required to agree to in order to proceed. Click the *Agree* button to continue without reading the license agreement (as is tradition). The next screen provides users with the choice to enter a license key (if they already purchased one), or the option to use the free 30-day trial period and forego entering a license key at this time. Select the *I have a license key for VMware Fusion 12* radio button and input the license key into the input box, or select the *I want to try VMware Fusion 12 Professional for 30 days* radio button, then click the *Continue*.

What about the *Get A Free License Key* button?

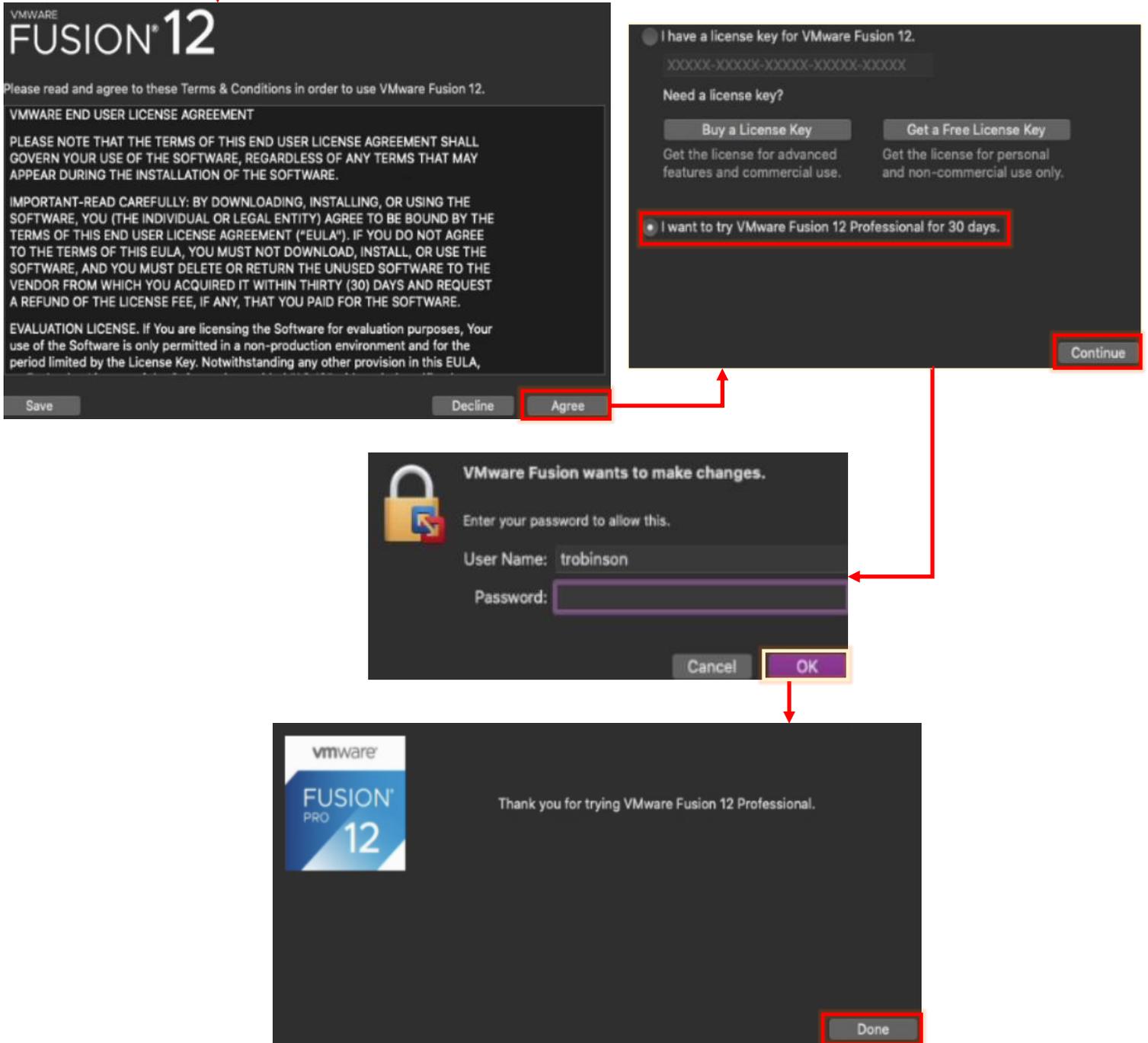
You may notice a button on this screen labeled *Get a Free License Key*. Like most things that appear to be too good to be true, it's a half-truth. Recently, VMware released a new product for MacOS users, Fusion Player. Technically, its free for personal and/or non-commercial use, but in layman's terms, it's the equivalent of VMware Workstation Player – you can run virtual machines with it, but not that many, and other bells and whistles are mostly locked behind paywalls. This includes the virtual network editor, which is absolutely required for setting up the lab network environment properly. Unfortunately, this means Fusion Player is not a viable substitute.

Students will be prompted to enter their username and password once more before being allowed to proceed. Afterwards, a new screen will pop up, thanking users for trying VMware Fusion 12 Professional. Click the *Done* button to complete the installation. Unfortunately, the fun is not over just yet.



11-2: As always, the software installation process on MacOS begins innocently enough. Double-click on the VMware Fusion .dmg file to mount the disk image. This causes a window to pop up asking students to double click the huge button to load the installer. From here, MacOS wants to remind you that the internet is full of irredeemable evil things. Click the *Open* to signify that yes, you downloaded this evil installer from the internet, and that you know what you're doing. Not yet content wasting your time, this prompts MacOS to ask for your username and password to run the VMware Fusion installer proper.

Continued from *fig. 11-2*



11-3: Welcome to the VMware Fusion Pro installer. The first screen contains the End User License Agreement. Click the *Agree* button to proceed without reading it (as is tradition). The next screen prompts users to enter a license key, or use the 30-day trial period. After you have made your choice by selecting the corresponding radio button (and/or entering a valid license key), click *Continue* to load yet another pop-up requesting the current user's password to proceed. After clicking *OK*, the installer continues, and students are met with a final screen, thanking them for trying VMware Fusion Professional. Click *Done* to exit the installer.

11.1.1.1 Permissions Dive

Note: MacOS has a really bad habit of telling users while they are using an application, that they need a specific system permission to perform a specific function. In most normal circumstances, an operating system would present users with an OK button to make those changes on the fly, so as not to interrupt your workflow. But as always, MacOS is different, not necessarily better. Instead, you get prompted to visit the *System Preferences* menu yourself, unlock it with your password, and make the indicated changes manually.

I'm telling you all of this because this "guess what permissions I need, then go and apply them yourself" is a fairly recent thing added in MacOS 10.15, and still prevalent in 11.x (Big Sur). If you're new to MacOS, you'll be seeing these permission pop-ups a lot, until all of the software you utilize on a regular basis have their required system permissions manually registered.

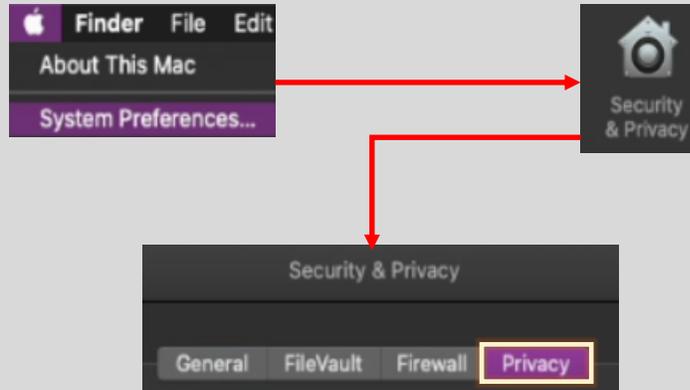
As soon as students click *Done*, two more dialogue boxes pop up. VMware Fusion will request access to the system's "Accessibility" features, and "System Events" feature. Click the *OK* button to allow VMware Fusion access to these system features. Another dialogue box will appear requesting users open the *System Preferences Privacy* menu, and assign the permissions themselves. Upon clicking the *Open System Preferences* button, the *System Preferences* menu is opened to the *Privacy* tab. The first thing students will need to do is click the large pad lock icon at the bottom of the window. This opens a dialogue box requesting their username and password in order to "unlock" the *Privacy* menu. With the menu unlocked, make the following changes:

- Click on *Accessibility*, then click the checkbox labeled *VMware Fusion*
- Click on *Automation*. There should be an entry labeled *VMware Fusion Applications Menu*. Click the checkbox labeled *System Events*
- Click on *Full Disk Access*. Click the checkbox labeled *VMware Fusion*. A pop-up will appear labeled "*VMware Fusion*" will not have full disk access until it is quit. Click the *Quit Now* button. This will close any running instances of VMware Fusion.

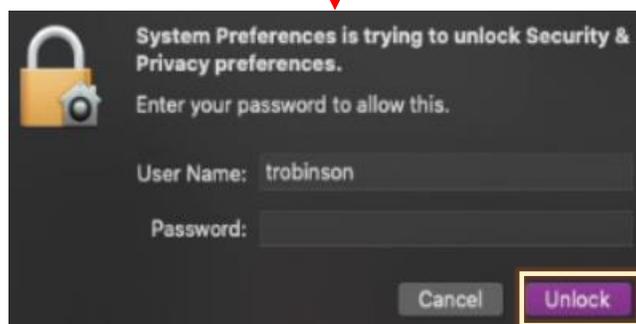
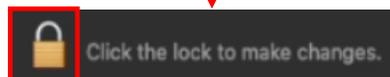
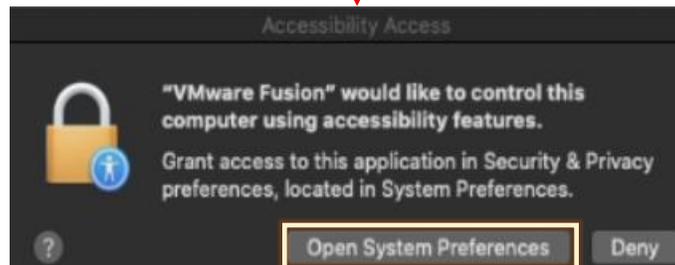
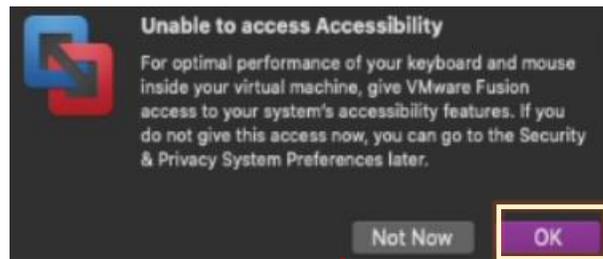
When students are finished making all of these changes, click the unlocked padlock icon to "lock" it again and prevent further changes from being made, then close the *System Preferences* menu.

Alternative Way to access *System Preferences Privacy Settings*

Did you get distracted or otherwise confused by the number of pop-up windows appear, asking for permissions to things left and right and did not click on the *Open System Preferences* button? Click the Apple icon in the upper left corner of the desktop, and select *System Preferences*. From there, click *Security & Privacy*, then click on the *Privacy* tab.

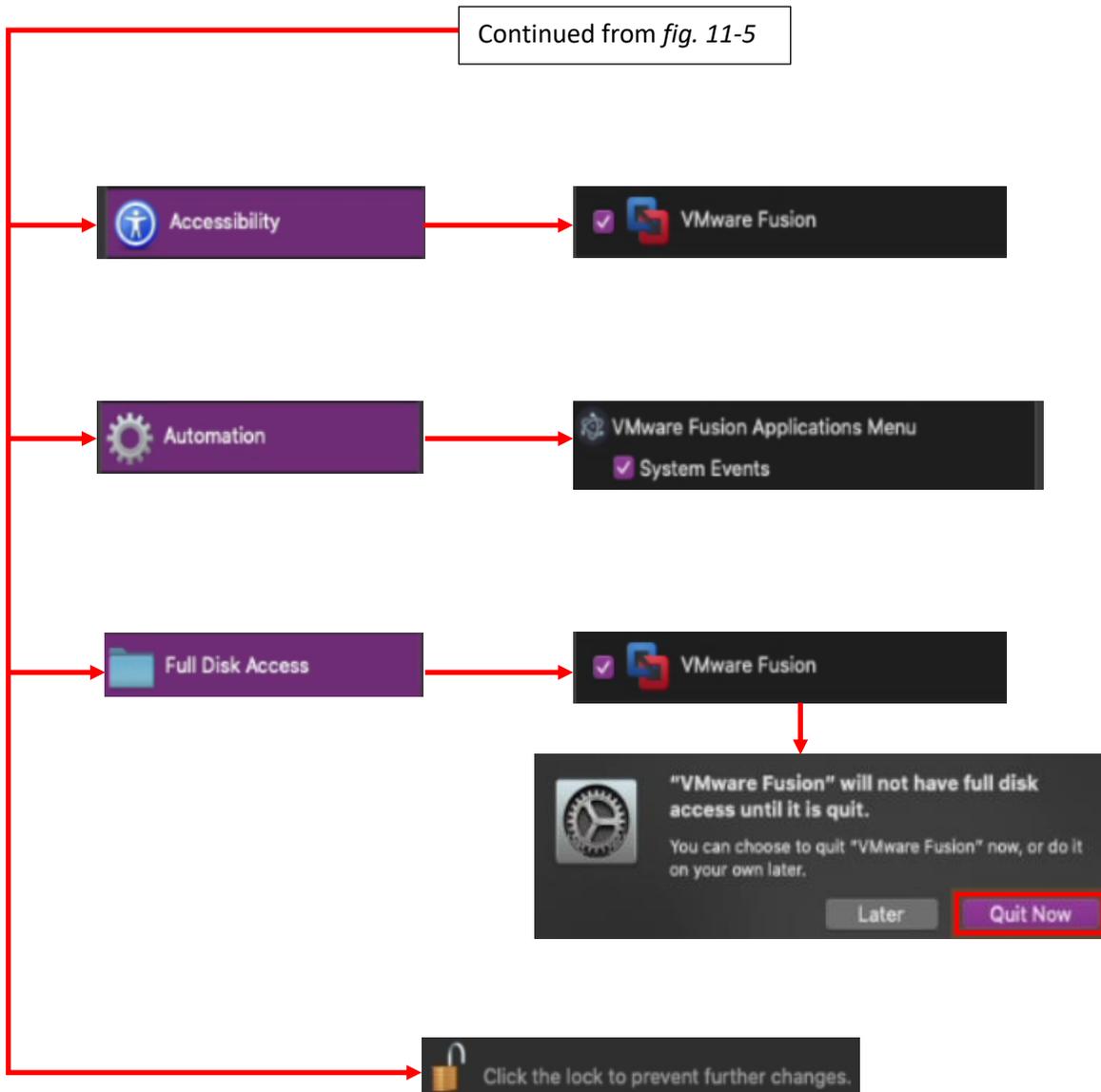


11-4: Did you accidentally click out of one of the many dialogue boxes asking for permissions, and now you need access to the *Privacy* settings in the *System Preferences* menu? Click the Apple icon and select *System Preferences* from the menu. Select the *Security & Privacy* option, then click on the *Privacy* tab to modify the settings above.



Continued to *fig. 11-6*

11-5: As soon as students click *Done* to finish the VMware Fusion installer, dialogue boxes will begin popping up asking for permissions to various system functions. When you click the *OK* button, another dialogue box pops up, more or less saying "Open system preferences and do it yourself". Click the *Open System Preferences* button to access the *System Preferences* menu – Specifically the *Privacy* settings. At the bottom of the menu window, click the padlock icon. This opens *yet another* window for students to enter their username and password in order to "unlock" the *Privacy* settings. After entering your username and password yet again, click the *Unlock* button to proceed.

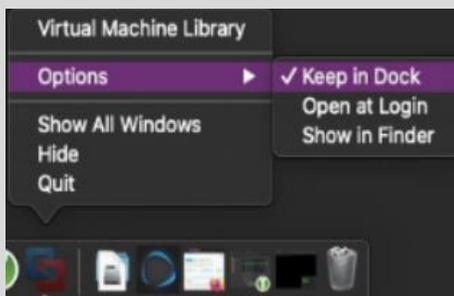


11-6: With *Security & Privacy* settings unlocked, students can finally make changes. We will need to grant VMware Fusion access to *Accessibility*, *Automation*, and *Full Disk Access* functions. To begin, click on the *Accessibility* menu option, then click on the checkbox labeled *VMware Fusion* to select it. Proceed to *Automation*, and click the checkbox labeled *System Events*. Finally, select *Full Disk Access*, and click the *VMware Fusion* checkbox there as well. When you do so, a dialog box appears notifying users that the VMware Fusion application will not get full disk access until the application is restarted. Click the *Quit Now* button to close any running copies of VMware Fusion. Once finished, click the unlocked padlock icon to "Lock" the *System Preferences* menu, then close the menu.

11.2 Virtual Network Editor

Open the VMware Fusion application. If students were not able to pin VMware Fusion to their dock, Open *Finder*, navigate to the *Applications* directory, locate the VMware Fusion icon, then double click on it.

Note: To bind VMware Fusion to the Dock, right click on the VMware Fusion icon, hover over *Options*, then click on the *Keep in Dock* setting.



11-7: I highly recommend binding VMware Fusion to the Dock for easy access.

When students first open VMware Fusion, the first thing they'll be greeted with is the *New Virtual Machine Wizard*, with the text *Select the Installation Method* in giant font. Students aren't quite ready to choose the form of the destroyer just yet, so ignore this window for now. Instead, click on the *VMware Fusion* listing in the navigation menu, then click *Preferences*. While there are a wide variety of different configuration options that can be modified here, we are interested in the *Network* settings tab. Click the icon labeled *Network*. At the bottom of the *Network* menu, click the padlock icon. In the dialogue box that appears, students should enter their username and password to "unlock" the network settings for customization.

With the network settings unlocked, students will need to create three virtual networks. To do this, click the small "+" icon below the left pane. Perform this task three times, to create the *vmnet2*, *vmnet3*, and *vmnet4* virtual networks. Left-click on the *vmnet2* listing on the left pane to bring up its settings. Locate the checkbox labeled *Connect the host Mac to this network* and ensure that it is checked. Immediately under that checkbox, locate the checkbox labeled *Provide address on this network via DHCP*, and uncheck it. *vmnet2* will act as the host-only (management) network for the lab network. **Do not use the Private to my Mac (vmnet1) network.**

Next, Left-click on *vmnet3*, locate both the *Connect the host Mac to this network* and *Provide address on this network via DHCP* checkboxes and uncheck them. Repeat this process once more for *vmnet4*. These two networks will act as the IPS1 and IPS 2 networks for the lab environment. Once finished, click the unlocked padlock icon at the bottom of the menu to lock the Network settings menu again, and close the *Preferences* menu.

Why aren't we using the *Private to my Mac* Network?

Some of you might be wondering why we aren't using the Private to my Mac network (vmnet1). That is because, for some reason or another, there is no way through the Network tab to actually disable DHCP for this network segment. We need to be able to disable's DHCP services, because we want our lab environment to get DHCP from the pfSense VM. Attempting to use the pfSense VM's DHCP service without disabling VMware Fusion's DHCP service will cause lab virtual machines to get different IP addresses and result in serious network problems.

Private to my Mac

The virtual machine is connected to your Mac using a private virtual network. The private network is not normally accessible from the physical networks on the Mac.

Multiple virtual machines can be connected to the same private network.

MTU: System Configuration

11-8: For reasons that are entirely beyond me, the only thing students can modify for the vmnet1 network is the Maximum Transmission Unit (MTU) size. You can't disable Fusion's DHCP service for this network. We want to use the DHCP service on the pfSense VM because its better in practically every way. For that reason, I made you create vmnet2, to serve as the host-only "management" network for our lab environment.

Promiscuous mode

You may have noticed the *Require authentication to enter promiscuous mode* checkbox at the bottom of the Network tab. In a nutshell, promiscuous mode is a special configuration for network interface cards that will let them capture packets that are not intended for them. It's required for the IPS VM to bridge the IPS1 and IPS2 networks together. With this configuration setting enabled, every time a network card wants to enter promiscuous mode and collect network traffic, you will need to enter your username and password to allow it. I recommended disabling it to avoid it becoming a nuisance.

With the padlock on the *Network* tab unlocked, uncheck the box, and a dialogue box will pop up warning that doing this is a risk. Click the *Proceed* button and the checkbox will disappear. Be aware that you may need to do this more than once. I had issues with the checkbox refusing to remain unchecked. If this happens to you, exit the *Preferences* menu, re-open it, and try again.

Require authentication to enter promiscuous mode



Promiscuous mode allows virtual machines to access and send additional data on networks connected to your Mac

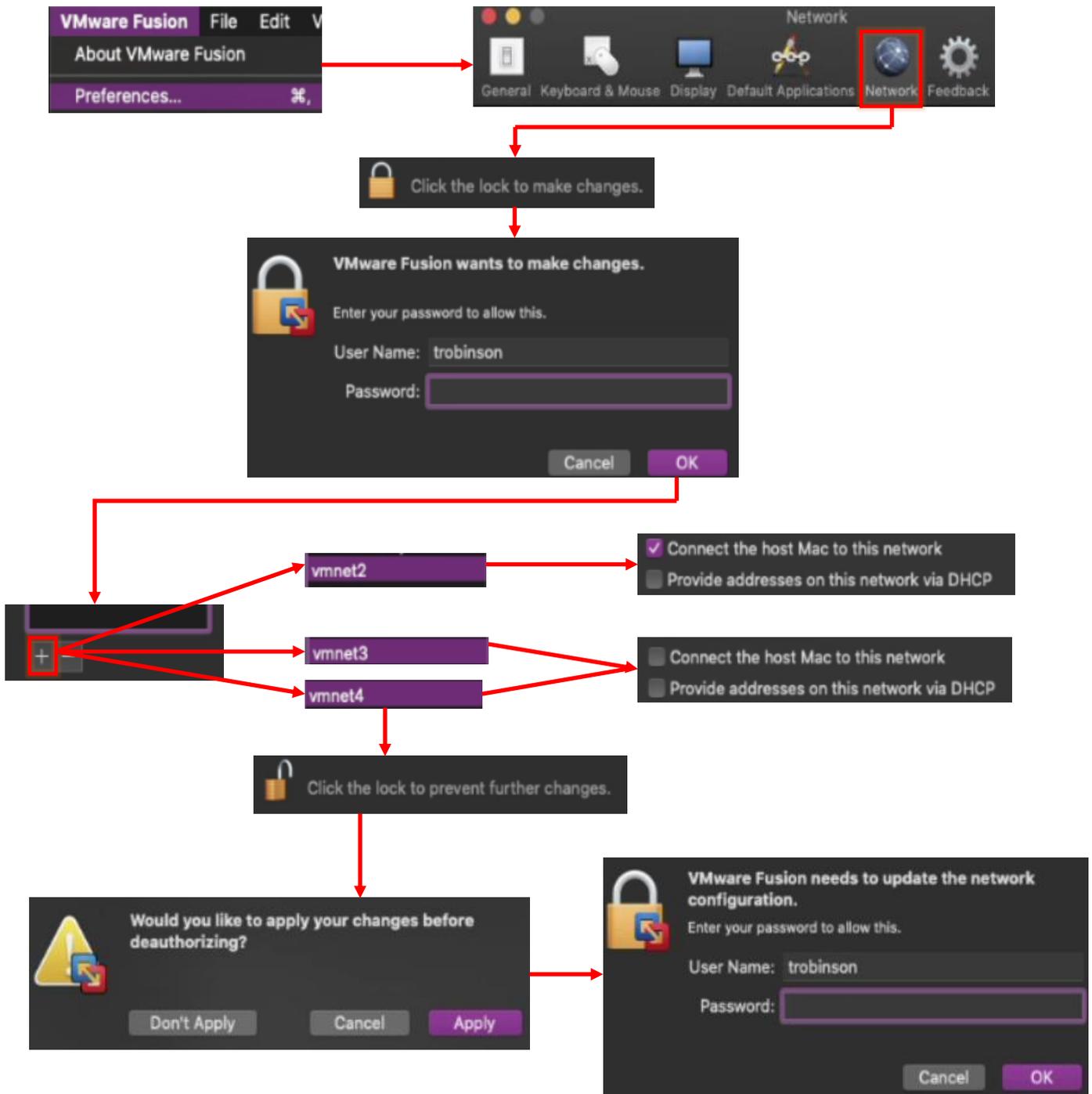
Disabling authentication allows all virtual machines to enter promiscuous mode without additional prompting. Promiscuous mode allows virtual machines to access all network packets and to send network packets not normally allowed. This setting should be enabled only when you trust the contents of all your virtual machines and all of the users of this Mac.

Cancel

Proceed

Require authentication to enter promiscuous mode

11-9: I recommend unchecking this configuration option. If you don't, every time you want to sniff/collect network traffic, you'll be prompted to enter your username and password. I also ran into a bug where the checkbox refused to remain unchecked. Exit the *Preferences* menu, re-open it, and try again if this happens to you.



11-10: Open the network settings menu by clicking *VMware Fusion > Preferences*. Click on the icon labeled *Network*, and at the bottom of the window, click the padlock icon. Enter your username and password in the pop-up that appears to continue. Under the left window pane, click the small "+" button three times. This will create three new virtual networks labeled *vmnet2*, *vmnet3*, and *vmnet4*. Click the *vmnet2* listing in the left pane, and in the right pane, Uncheck the *Provide addresses on this network via DHCP* checkbox. Next, click on *vmnet3*, and uncheck both the *Provide addresses on this network via DHCP* and *Connect the host Mac to this network* checkboxes. Repeat this process for *vmnet4*. Once finished, click the unlocked padlock icon, then click the *Apply* button in the pop-up window that appears. This causes one final window to appear, asking students for their username and password. After authenticating once again, students may close the menu window.

11.3 Configuring the vmnet2 Host Virtual Adapter

Before students can continue, they will need to configure the vmnet2 host virtual adapter with an IP address. Open up the *iTerm* or *iTerm2* terminal application, and enter the commands:

```
ifconfig vmnet2
sudo ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
ifconfig vmnet2
```

The first `ifconfig` command is to confirm that the `vmnet2` interface exists. The second command uses `sudo` to gain root permissions in order to use the `ifconfig` command to manually configure an IP address and subnet mask for `vmnet2`. Finally, the third command (exactly the same as the first) is used to verify that the correct IP address and subnet mask has been applied. The `inet` field should contain the value `172.16.1.2`, and `172.16.1.255` for the broadcast field.

vmnet2 and its IP address settings have disappeared. Why?

Students may have noticed that upon rebooting their host operating system, `vmnet2` and any interface configurations performed (like, say setting the IP address and subnet mask) are gone. Starting VMware Fusion will at least bring back `vmnet2`, but the IP address configuration is still gone. Unfortunately, this is something of a known issue with some hypervisors on MacOS – ***this is not a VMware Fusion-only problem.***

Unfortunately, after having done this for years, I am still unaware of any method of getting the `vmnet2` interface itself, and/or any customizations performed against the interface to persist between reboots on MacOS. The only method that seems to work would be to restart Fusion, and reperform the steps necessary to set the IP address and subnet mask for `vmnet2`.

Fortunately, I've created a simple script available for MacOS users that automates this process, called Flightcheck-OSX. You'll learn more about it in Chapter 15. For now, I ask readers to bear in mind that every time the host system reboots, they will be required to restart their VMware Fusion, and reconfigure the IP address of `vmnet2` manually.

```
trobinson@trobinsons-MacBook-Pro ~ %1ifconfig vmnet2
vmnet2: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
trobinson@trobinsons-MacBook-Pro ~ %2sudo ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
Password:
trobinson@trobinsons-MacBook-Pro ~ %3ifconfig vmnet2
vmnet2: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:50:56:c0:00:02
inet 172.16.1.2 netmask 0xfffff00 broadcast 172.16.1.255
```

11-11: `ifconfig` is ran the first time (1) to make sure `vmnet2` is present on the system, the second time (2) with `sudo` in order to change the IP address and subnet mask of `vmnet2`, and the third time (3) to confirm the IP address and netmask has been applied correctly via the `inet` and `broadcast` fields.

11.4 Building the first Virtual Machine, pfSense

The pfSense virtual machine is responsible for binding the entire lab environment together. It is a well-supported firewall distribution with amazing ease of use and functionality. pfSense is also very modular, featuring a system for adding on additional functionality through BSD's pkg software package manager.

It is recommended for students to download all of the ISOs (e.g., pfSense, Ubuntu Server and Kali Linux), and pre-built virtual machines (e.g., Metasploitable 2) required for their lab environment in advance. Check out chapter 1, [section 1.5.4](#) (p. 26) for download links. Additionally, students must decompress the pfSense installation ISO before attempting to boot from it. [Section 1.8](#) (pp. 33-35) covers how to perform this task.

11.4.1 VM Creation

VMware Fusion virtual machines are created through a step-by-step process called the *New Virtual Machine Wizard*. Open the VMware Fusion application. In the navigation bar, select *File > New* to begin the wizard. Alternatively, students can click the + icon in the virtual machine library window, then click *New* in the drop-down menu that appears to access the wizard as well.

The first screen in the wizard is labeled *Select the Installation Method*. Select the *Create a custom virtual machine* option, then click the *Continue* button to proceed. The next screen, titled *Choose Operating System* has students select the operating system they want to install. Select *Other > FreeBSD version 10 and earlier 64-bit*, then click *Continue*.

Note: pfSense CE is based on FreeBSD. If you're reading this in the future, Netgate (the makers of pfSense) has a support page that details what version of FreeBSD each version of pfSense is based on here:

<https://docs.netgate.com/pfsense/en/latest/releases/versions-of-pfsense-and-freebsd.html>

On the *Choose Firmware Type* screen, select the *Legacy BIOS* radio button, then click *Continue*. On the next screen, *Choose a Virtual Disk*, select the *Create a new virtual disk* radio button, then click *Continue* once more.

Note: If you're saying to yourself, "I thought we only needed 5GB of space for the pfSense VM. What gives?" We'll be fixing this momentarily.

Finally, the *Finish* screen. Click the *Customize Settings* button, because there are a lot of changes that need to be made. Upon clicking this button, a window appears, asking students to name their virtual machine and choose where they would like to store their virtual machine. In the *Save As* input box, enter pfSense. In the *Where* drop-down menu, accept the default location of the *Virtual Machines* directory.

Note: Unlike every other hypervisor out there, you don't have the option of setting a new default virtual machine directory. Also, VMware Fusion isn't exactly forthcoming with where the "Virtual Machines" directory is actually located. The full path is:

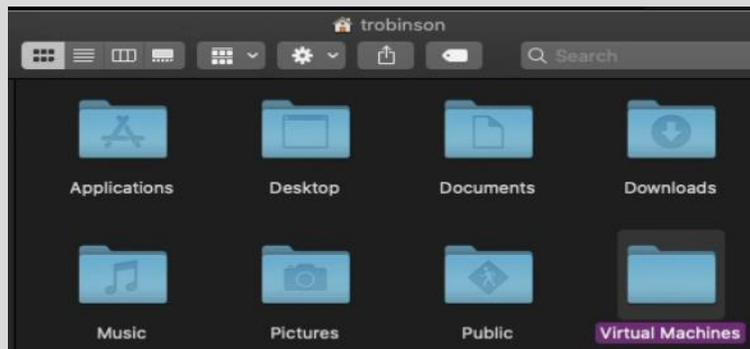
```
/Users/[username]/Virtual Machines.localized
```

Replace the [username] placeholder with your username. For example:

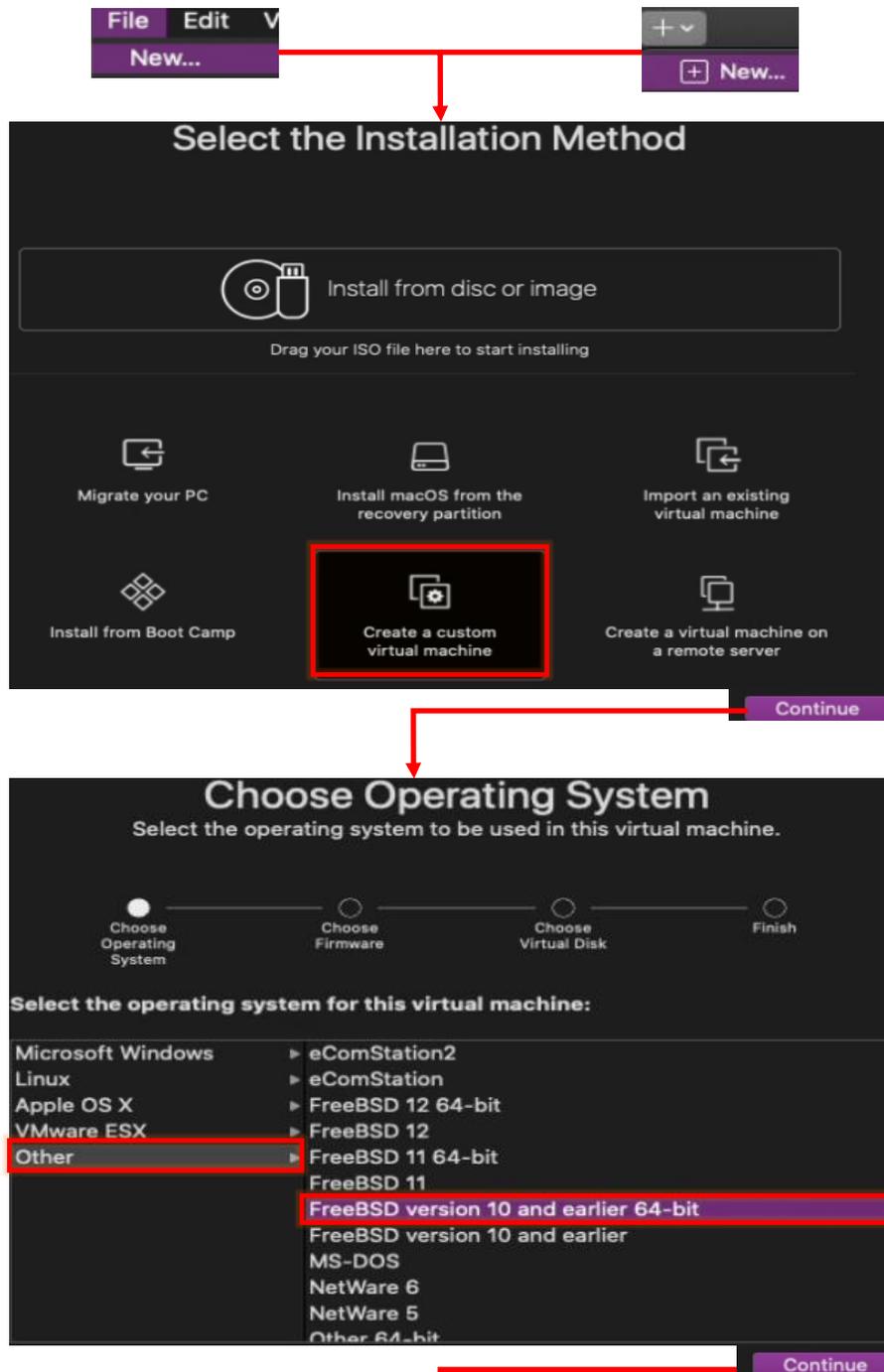
```
/Users/trobinson/Virtual Machines.localized
```

But if you're trying to find this directory with *Finder*, look under your user's home directory for the folder labeled *Virtual Machines*.

```
trobinson@trobinsons-MacBook-Pro Virtual Machines.localized % pwd
/Users/trobinson/Virtual Machines.localized
```



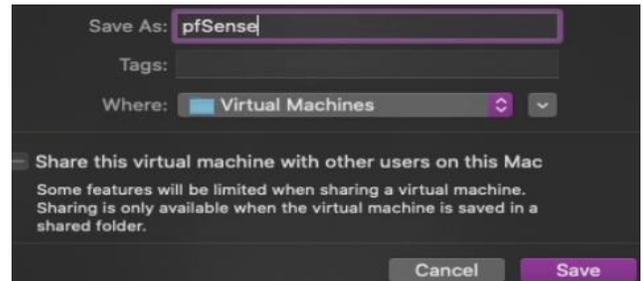
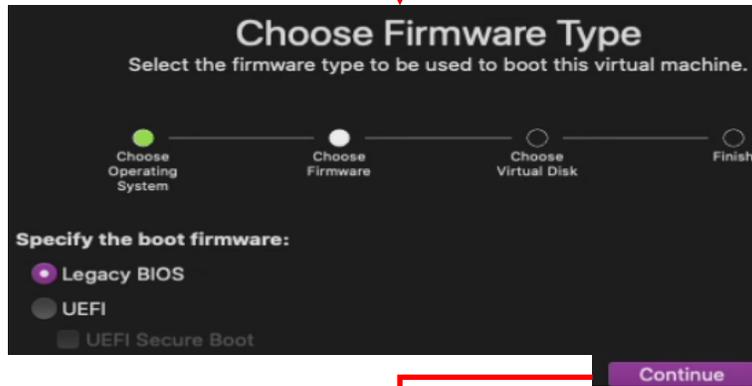
11-12: VMware Fusion stores virtual machine files in the `Virtual Machines.localized` folder within your user's home directory. But it'll be displayed as just the *Virtual Machines* directory in *Finder*.



Continued to *fig. 11-14*

11-13: Start the new virtual machine wizard by selecting *File > New* from the navigation menu, or by clicking the "+" button in the virtual machine library window, followed by *New* in the drop-down menu. The first screen is labeled *Select the Installation Method*, and should be familiar – it's the first screen students observed after running VMware Fusion for the first time. Click the *Create a custom virtual machine* option to highlight it, then click *Continue*. On the *Choose Operating System* screen, select *Other > FreeBSD version 10 and earlier 64-bit* then click *Continue*.

Continued from *fig. 11-13*



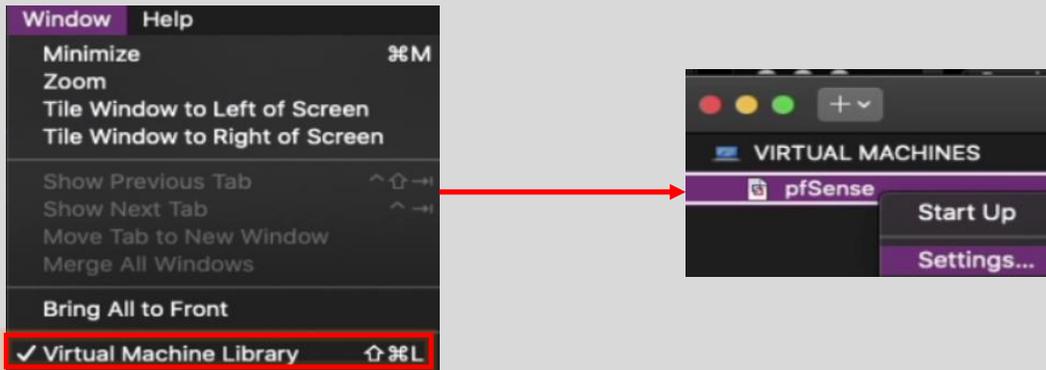
11-14: On the *Choose Firmware Type* screen, select the *Legacy BIOS* radio button, then click *Continue*. This brings students the *Choose a Virtual Disk* screen. The default selection should be the *Create a new virtual disk* option, so all students should need to do is click *Continue* to proceed to the *Finish* screen. In spite of its name, we're nowhere near done configuring this VM. Click the *Customize Settings* button. This will cause a pop-up window to appear. In the *Save As* input box, enter *pfSense*, then click *Save*.

11.4.2 Customizing the pfSense VM

If students clicked the *Customize Settings* button from the new virtual machine wizard, a new window labeled *pfSense: Settings* will appear, with a host of icons for all of the settings and virtual hardware students can modify. Please note, that students can click the *Show All* button from any of the submenus to return back to the main menu.

Jumped the Gun

If you accidentally clicked *Finish* at the end of the new virtual machine wizard, or accidentally closed the *pfSense: Settings* menu, open the *Virtual Machine Library*. This is the window should automatically appear at the end of the new virtual machine wizard. It can also be accessed from the navigation menu, under *Window > Virtual Machine Library*. On the left pane, under the text *VIRTUAL MACHINES*, is a listing of all virtual machines VMware Fusion is currently aware of. Right click on the pfSense entry and select *Settings*.



11-15: If you need to access the virtual machine settings menu, locate the target virtual machine in the *Virtual Machine Library* window (which can be opened from the navigation menu under *Window > Virtual Machine Library*), then select *Settings*.

Change the following settings:

- Click the *Sharing* and ensure that the *Enable Shared Folders* checkbox is **unchecked**
- Click on *Processors & Memory*. In the memory input box, enter 512(MB) for the amount of RAM to allocate to the pfSense VM.
- Click on *USB & Bluetooth*. In small text under the *Connect USB devices* pane is the text *Advanced USB options*. Click on that text, and additional options appear. Click the *Remove USB Controller* button and then click *Remove* in the pop-up window that appears
- Click on *Sound Card*, then click on the *Remove Sound Card* button. Once again, Click the *Remove* button in the pop-up that appears.
- Click on *Camera*, then click the *Remove Camera* button. A pop-up will appear asking if students are sure they want to remove the camera. Click the *Remove* button again to assert your dominance to the machine legion.

Note: MacOS may present a pop-up window stating that "VMware Fusion" would like access to the camera. Seeing as how we are here to remove the thing, click *Don't Allow*.

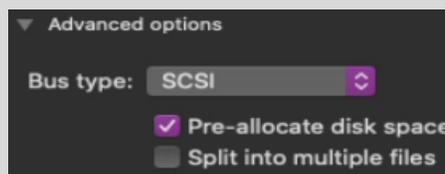


11-16: No thanks.

- Click on *CD/DVD (IDE)*. Check the checkbox next to *Connect CD/DVD Drive*. In the drop-down menu below labeled, *This CD/DVD drive is configured to use the following*, select the *Choose a disc or disc image* option. This will open the file browser on your Mac. Navigate to the location of the decompressed pfSense ISO, and select it.
- Click on *Hard Disk (SCSI)*. Using either the *Disk size* slider, or the input box all the over to the right, enter 5.00 (GB) as the disk size, and click the *Apply* button in the lower right.

The Basics of Close Quarters Storage

Some of the more adventurous of you may notice that when you click on *Advanced options* in the *Hard Disk* sub-menu, additional options for pre-allocating disk space, and how the VMDK file is stored on disk are made available. Many of you using VMware Fusion are probably going to be using it on a MacBook/MacBook Pro with very limited space, with a significant portion of that already allocated to MacOS, and a handful of applications. Pre-allocating disk space when you don't actually have is a terrible idea. If you happen to be running Fusion on Apple hardware that actually has a decent amount of storage, then go ahead and check *Pre-allocate disk space*, and uncheck *Split into multiple files*. Generally, these configuration options improve virtual machine disk performance, at the cost of requiring all of the allocated disk space up front. Keep this in mind for later in the chapter when you create the SIEM, IPS, Kali and Metasploitable 2 VMs.

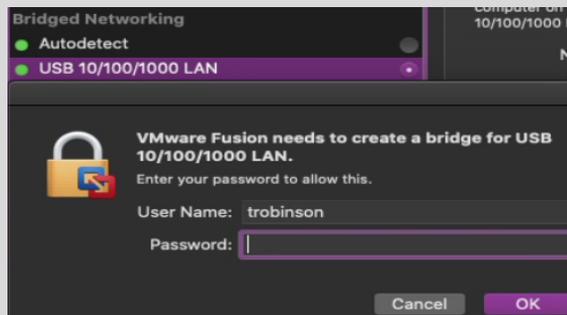


11-17: Some of you might have taken a peek at the *Advanced options* and noticed these settings. The reason why I'm not advising you to make these configuration changes is because some MacBooks don't have a lot of SSD storage at all, and MacOS plus a small collection of applications can take up a lot of disk space, easily. A 250GB SSD cannot sustain pre-allocating disk space for all of our lab VMs. If you have the storage space to support doing this however, you absolutely should take advantage of these options for the better VM performance.

- Click on *Isolation*. Uncheck both the *Enable Drag and Drop* and *Enable Copy and Paste* checkboxes.
- Click on *Network Adapter*. Ensure the *Connect Network Adapter* checkbox is selected. In the pane below, under the *Bridged Networking* section, select the *Autodetect* radio button. If students

have a particular network interface they wish to have the pfSense VM bridge to, they may choose that interface instead. Clicked the Advanced options text below the pane to display additional options. Click the *Generate* button and document the contents that appear in the MAC address input box.

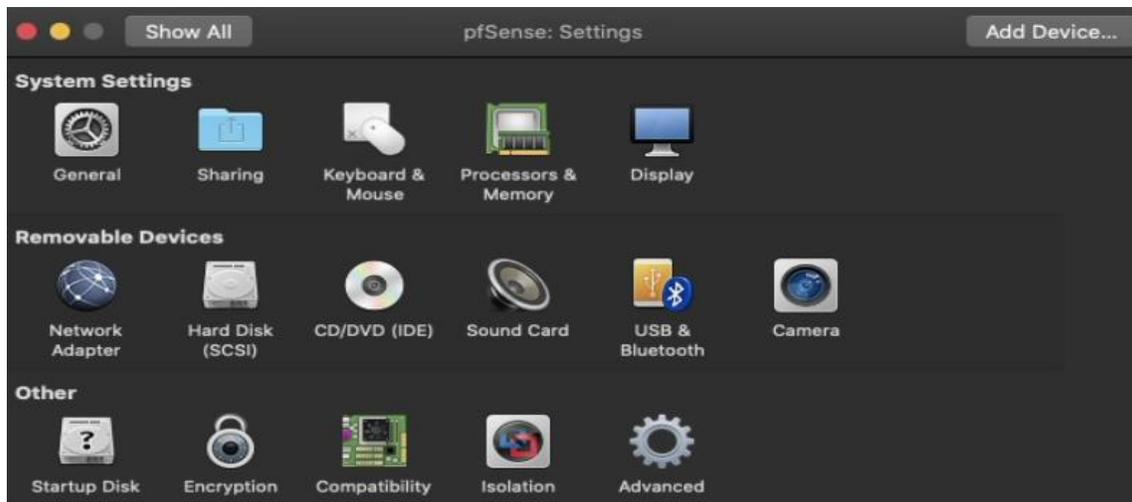
Note: When choosing a specific interface to bridge against, MacOS may pop up yet another dialogue box, requesting a username and password to authorize the creation of a network bridge on that interface.



11-18: When selecting a network interface to bridge to, MacOS may request your username and password to authorize the configuration change.

- Click on the *Add Device* button in the upper right corner of the menu window. Click the Network Adapter icon to highlight it, then click the *Add* button in the lower right corner. A new window appears, *pfSense: Network Adapter 2*. Ensure the *Connect Network Adapter* checkbox is selected, then select the *vmnet2* radio button in the listing below, under *Custom*. Once again, click *Advanced options*, then click the *Generate* button to generate a MAC address for *Network Adapter 2*. Document this MAC address as well.
- Repeat the *Add Device* process once more. Select *Network Adapter*, then click *Add*. In the *pfSense: Network Adapter 3* window, Ensure the *Connect Network Adapter* checkbox is selected, then select the *vmnet3* radio button in the listing below, under *Custom*. Click *Advanced options*, followed by *Generate* one last time, and record the MAC address generated for *Network Adapter 3*.

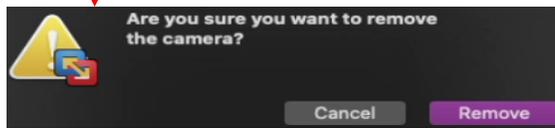
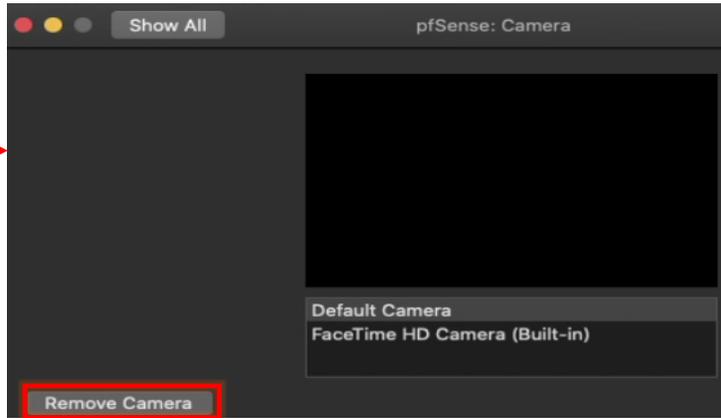
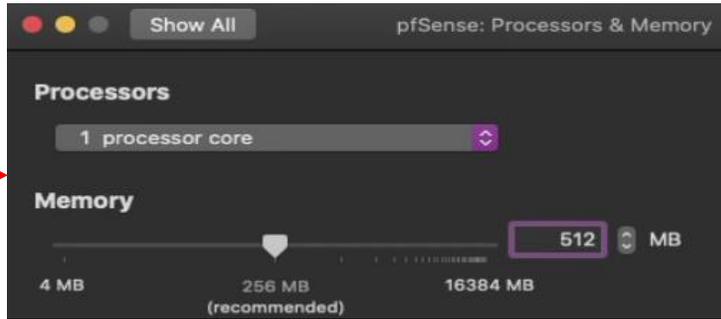
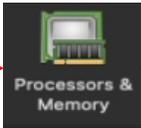
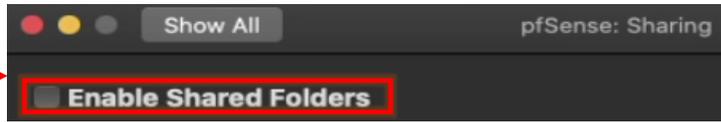
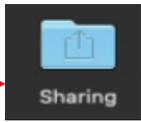
When students are finished editing these settings, close the *pfSense: Settings* menu.



Continued to *fig. 11-20*

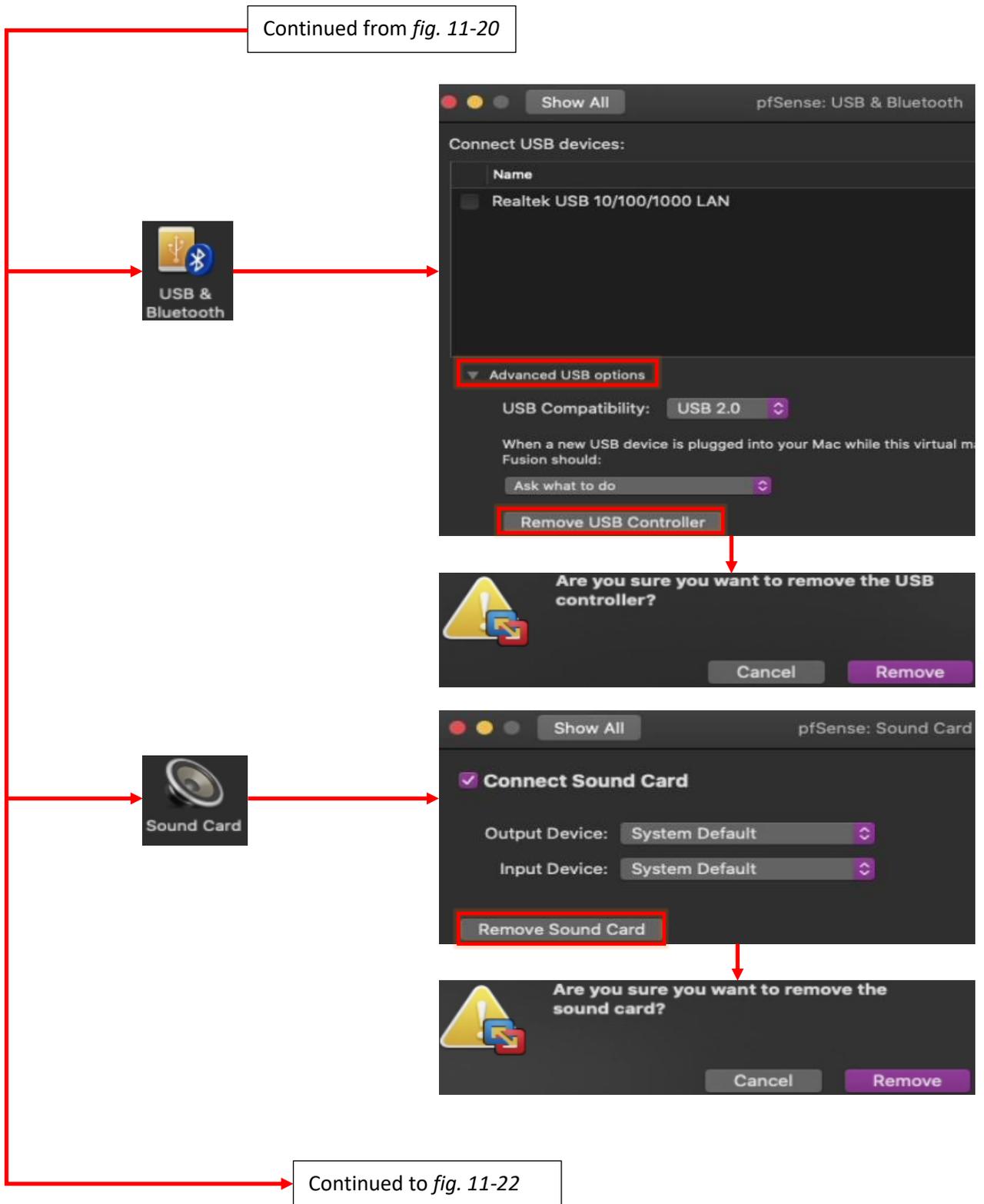
11-19: Welcome to the virtual machine settings menu. There is a lot of work for us to do here, because VMware Fusion makes a lot of assumptions when it creates your virtual machines (and of course, they are wrong). If you get lost navigating the sub-menus and settings here, the *Show All* button in the upper left will return you here.

Continued from *fig. 11-19*

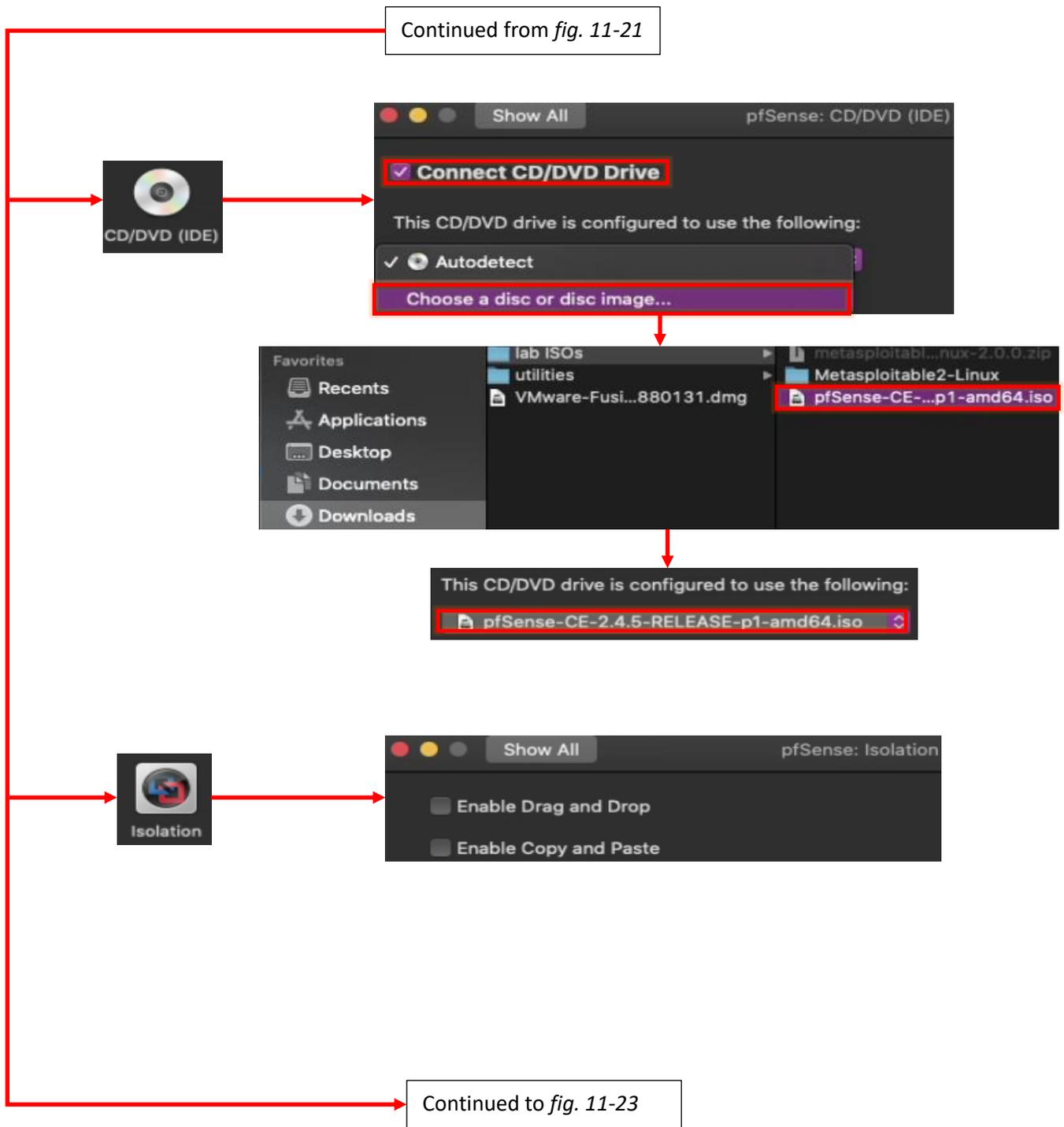


Continued to *fig. 11-21*

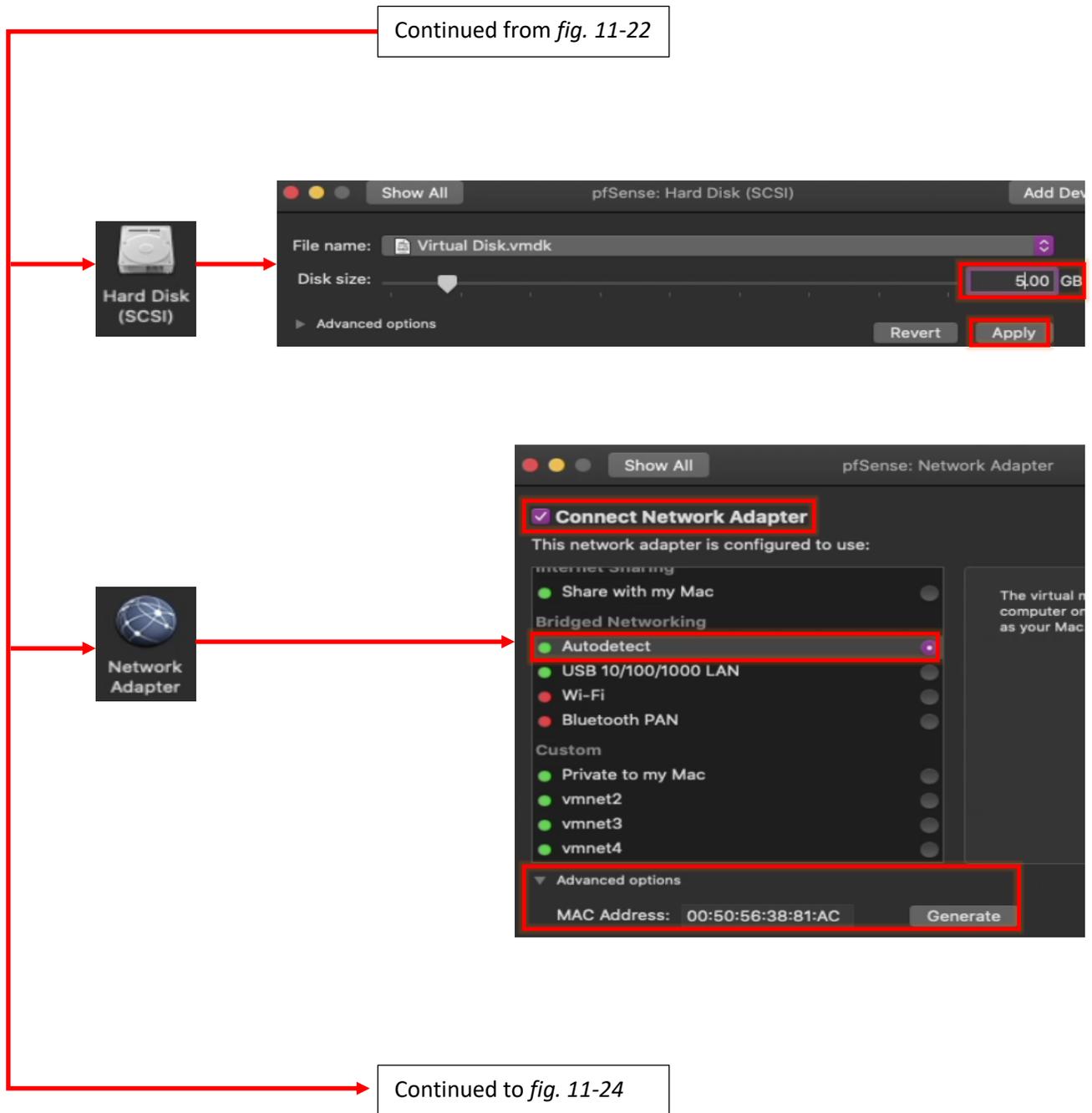
11-20: Begin by navigating to the *Sharing* menu, and ensure that the *Enable Shared Folders* checkbox is unchecked. Next, go to *Processors & Memory* and increase the amount of allocated memory to 512MB. Locate the *Camera* submenu, then click the *Remove Camera* button. A pop-up window appears asking if students are sure they want to do this. Click the *Remove* button to assert dominance over the machine legion.



11-21: Navigate to *USB & Bluetooth*, click *Advanced USB options*, followed by *Remove USB Controller*. Then, click the *Remove* button in the window that appears. Select *Sound Card*, then click the *Remove Sound Card* button, followed by the *Remove* button on the pop-up that appears.

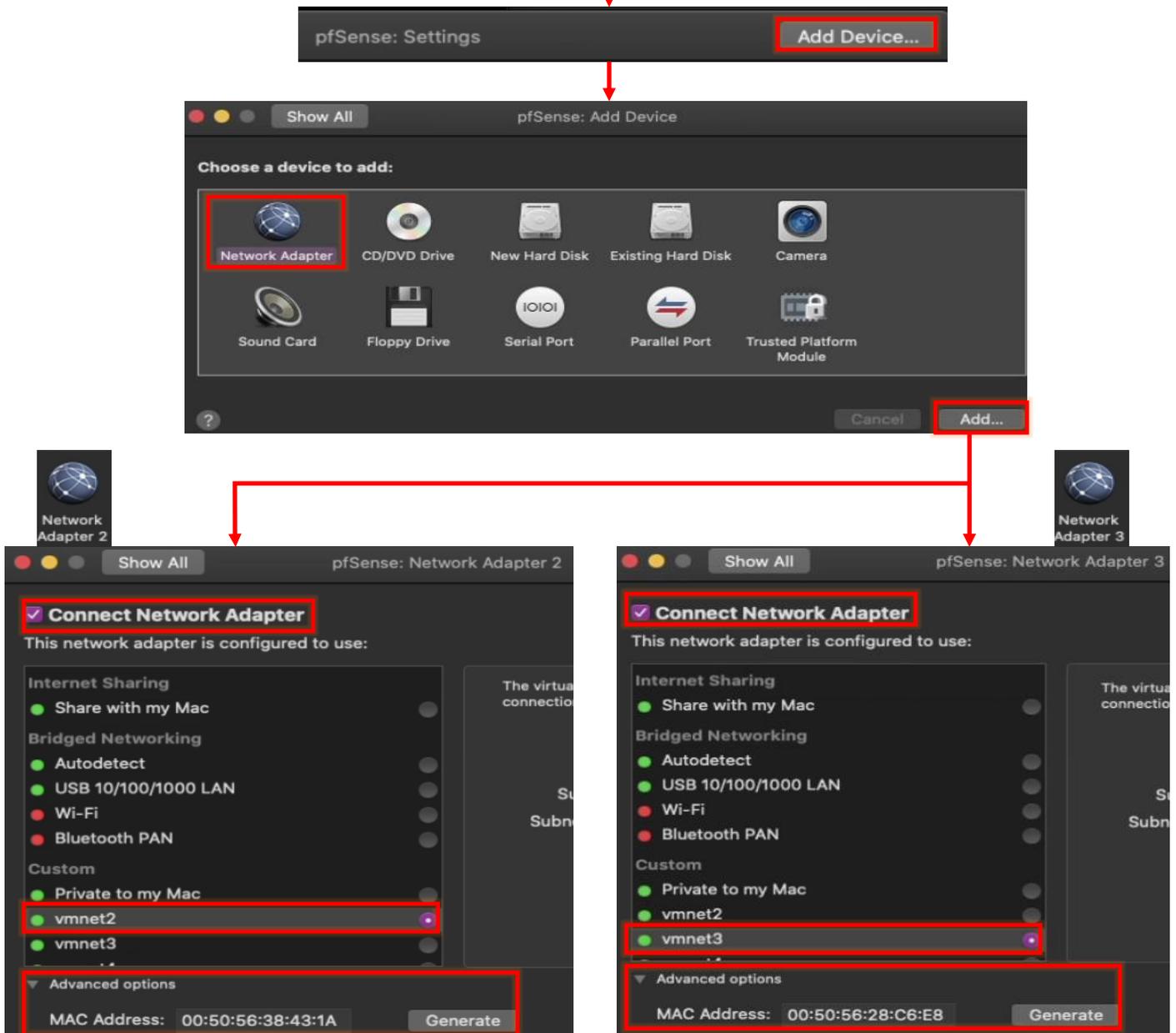


11-22: Select *CD/DVD (IDE)*. Begin by checking the *Connect CD/DVD Drive* checkbox. In the dropdown menu labeled *This CD/DVD drive is configured to use the following*, select the *Choose a disc or disc image* option. This causes a *Finder* window to open. Navigate to the location of the decompressed pfSense ISO and select it. Next, select *Isolation*, and uncheck both the *Enable Drag and Drop* and the *Enable Copy and Paste* checkboxes.



11-23: Click on *Hard Disk (SCSI)* and using either the slider or the input box on the right, change the *Disk Size* to 5.00GB, then click *Apply*. Click on *Network Adapter*. Ensure that the *Connect Network Adapter* checkbox is checked. In the pane below, under *Bridged Networking*, select the *Autodetect* radio button. Underneath the pane, click *Advanced options* to cause additional settings to appear. Click the *Generate* button to the right of the *MAC Address* input box, and record the contents that appear.

Continued from *fig. 11-23*



11-24: Last but not least, students need to add two more network adapters to the pfSense VM and configure them. Click the *Add Device* button in the upper right corner of the menu. A new window appears labeled *pfSense: Add Device*. Click *Network Adapter*, then click the *Add* button. Students are immediately directed to a new window *pfSense: Network Adapter 2*. Ensure that the checkbox *Connect Network Adapter* is selected, and in the pane below, select the *vmnet2* network under *Custom*. Underneath the pane, click *Advanced options*, then click the *Generate* button to the right of the *MAC Address* input box, and record the contents. Repeat this process one more time to create a third and final network adapter (*pfSense: Network Adapter 3*). Attach this network adapter to the *vmnet3* network, click *Advanced options*, *Generate* a MAC address, and copy the results down for later. Please note that the *pfSense: Network Adapter 2*, and *pfSense: Network Adapter 3* menus can be accessed at any time by clicking the *Network Adapter 2* or *Network Adapter 3* icon from the main *pfSense: Settings* menu. When students are finished, close the *pfSense: Settings* menu.

Noting the Notable

I can't overstate the value of documenting your lab network properly. Use whatever note-taking methods you prefer – paper and pen, Evernote, text editors, personal wikis, databases, spreadsheets, etc. Document the name of the VM, Operating system, the number of CPU cores allocated, RAM, Disk size, number of network adapters, network segments they are attached to, and their MAC addresses. This is called *asset management*, and it's an important habit to cultivate. Here is a template you can use for documenting your VMs:

VM Name:
Operating System:
CPU Cores:
RAM:
Disk Size:
Virtual Network Adapters:
Network Adapter #:
-Network Segment:
-MAC Address:
<Repeat for each network adapter>
Additional Notes:

And as an example, here is my pfSense VM entry:

VM Name: pfSense
Operating System: pfSense (FreeBSD)
CPU Cores: 1
RAM: 512MB
Disk Size: 5GB
Virtual Network Adapters: 3
Network Adapter 1:
-Network Segment: Bridged/WAN
-MAC Address: 00:50:56:38:81:AC
Network Adapter 2:
-Network Segment: Management/vmnet2
-MAC Address: 00:50:56:38:43:1A
Network Adapter 3:
-Network Segment: IPS 1/vmnet3
MAC Address: 00:50:56:28:C6:E8
Additional Notes: Lab firewall. Provides NTP, DNS, DHCP,
and HTTP proxy services.

Do this for every single virtual machine you add to your lab environment. Keep track of systems added or removed from the lab network. Always be aware of what's running on your networks. If you can do these things, you'll be better at asset management than most of the Fortune 500.

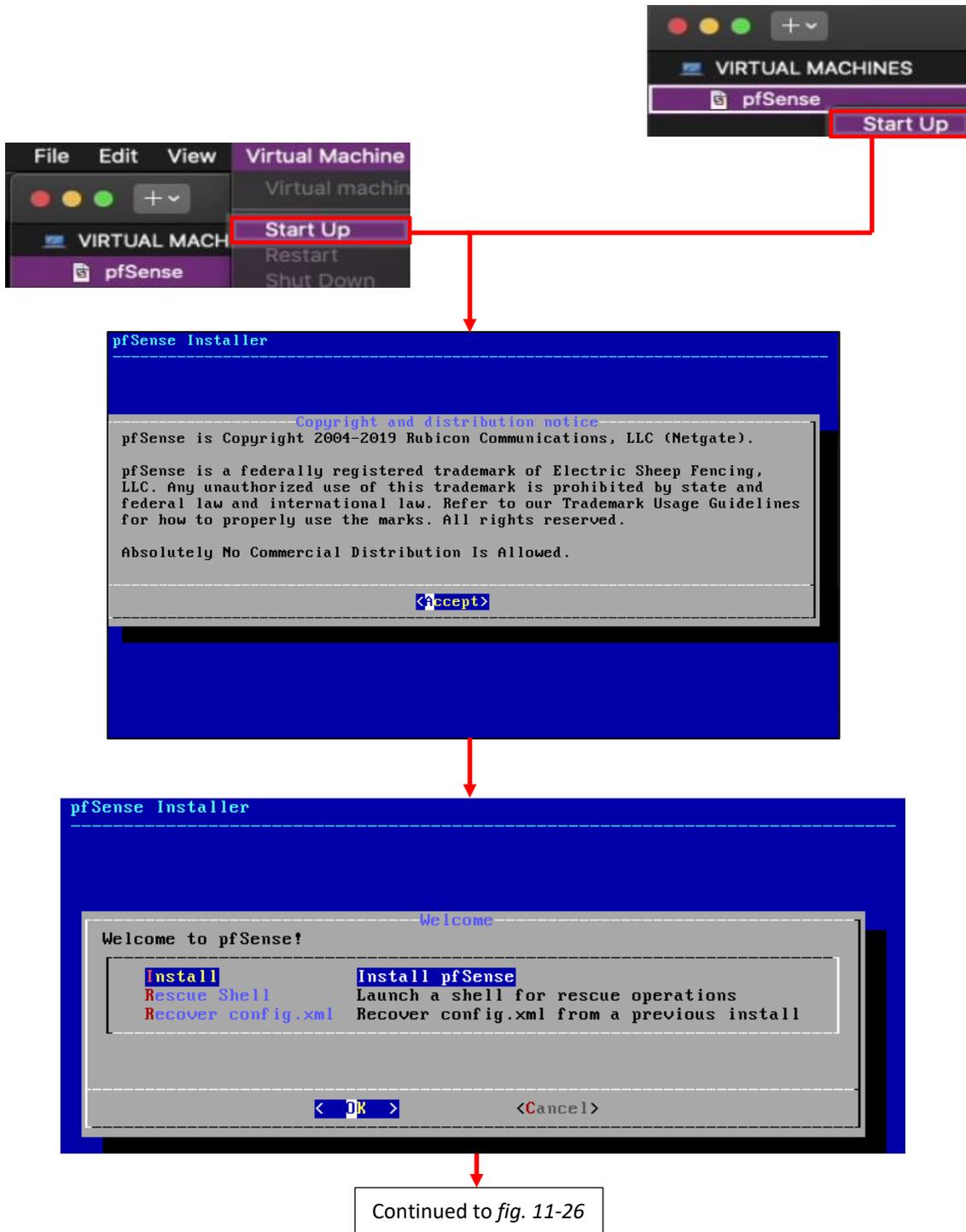
11.4.3 First Boot and OS Installation

Our virtual machine has been created, and it is now time to install the pfSense operating system to the new VM. To begin, students will need to power on the virtual machine, then connect to its virtual console. The easiest way to do this is to left-click on the pfSense entry in the *Virtual Machine Library* window to highlight it, right-click on the pfSense entry, then select *Start Up*. Alternatively, students can left-click the pfSense VM entry to highlight it, then select *Virtual Machine > Start Up* from the navigation menu.

The virtual console should pop up automatically. Think of this window as a direct keyboard, video, and mouse connection to the virtual machine while it is running. You'll notice a lot of text flying by as the VM boots from the installation ISO. Eventually you will reach the pfSense Installer. The first screen shows the *Copyright and distribution notice* for the software. Click anywhere in the virtual console window, and hit Enter to accept the software terms and conditions

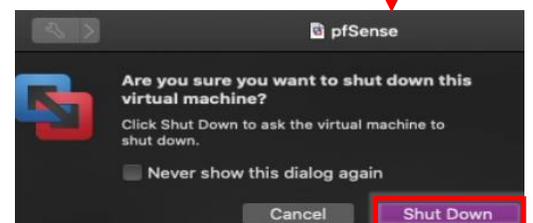
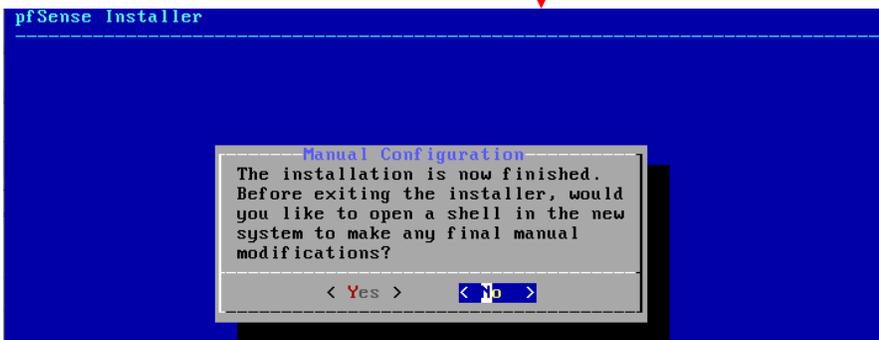
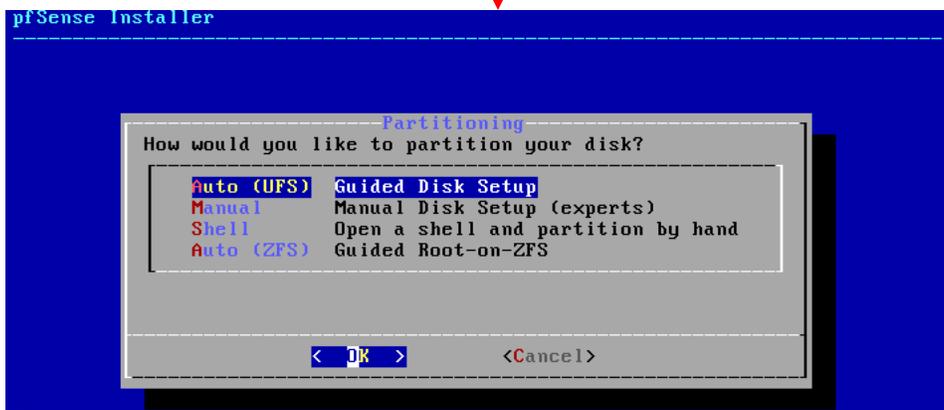
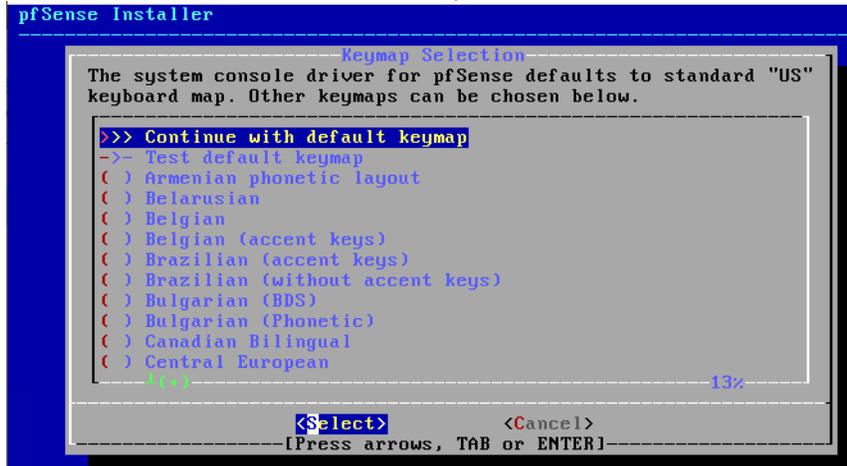
Next is the *Welcome* screen for the OS Installer. The option *Install pfSense* should be highlighted by default, but if not, use the arrow keys on your keyboard to select it, then hit enter. The next screen, titled *Keymap Selection* appears. If students are from a region of the world with a unique keyboard layout, they will need to search for and select it. Otherwise, select *Continue with default keymap* to use the US keymap, and hit enter. Next is the *Partitioning* screen. Partitioning is used to tell the installer how much and what portion of the disk to allocate. Since this is a virtual machine, and the disk is relatively tiny (5GB), select *Auto (UFS) Guided Desk Setup* and press enter to tell the installer to use all of the available disk space.

The installer handles formatting the disk and copying the operating system files over. The next screen, titled *Manual Configuration* asks if students want a command shell to manually edit any operating system files before closing the installer. Select *No*, and hit enter again. Finally, on the *Complete* screen, select the *Reboot* option and hit enter. While the VM is rebooting, hit the Ctrl+Command (or Ctrl+Meta) keyboard combination to release control of the keyboard and mouse back to the host operating system, left-click on the pfSense VM entry in the *Virtual Machine Library* pane once more, then select *Virtual Machine > Shut Down* from the navigation menu. VMware Fusion will ask students to confirm if they wish to power off the virtual machine. Click the *Shut Down* button to proceed (and optionally the *Do not show this message again* checkbox).



11-25: Power on the newly created pfSense VM. VMware workstation should automatically connect students to the virtual console. The VM should boot from the installation ISO automatically. Accept the License Agreement, then select the *Install pfSense* option to proceed.

Continued from *fig. 11-25*



11-26: The rest of the installation process is pretty straightforward. Most students will be able to hit enter the entire way through and accept the defaults. Upon reaching the *Complete* screen, select the option to Reboot the virtual machine, and while the VM is rebooting, shut it down.

Virtual Machines Ate My Neighbors Input

When a user clicks on the virtual machine console, that window grabs all of the input from the mouse and keyboard. That means that if there are other applications running on your host system you want to interact with, you have to tell the virtual console to let go of the mouse and keyboard first. VMware Workstation uses a special key binding called the *Hot key combination* to signal to the virtual console that you have other things to do. On Windows and Linux, this is the Ctrl+Command (or on other keyboards the Ctrl+Meta, or Ctrl+Windows) key combination. This keyboard combination can be found and modified in *VMware Fusion > Preferences > Keyboard and Mouse*. Once there, click on the *Fusion Shortcuts* tab.

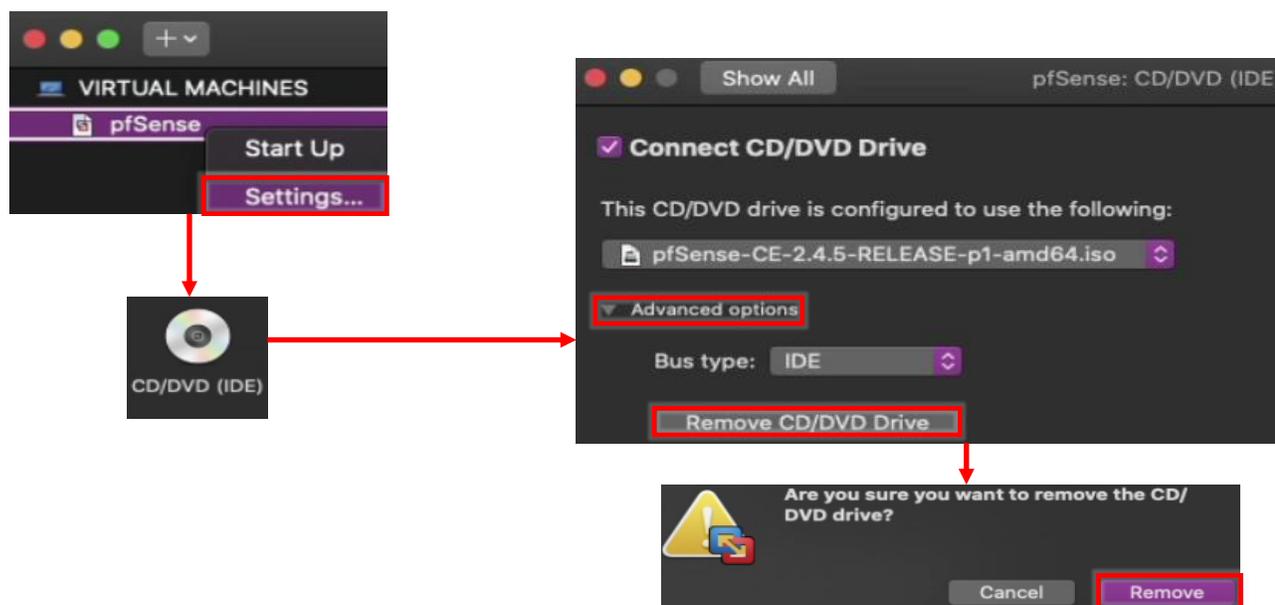


11-27: Look for the *Hot key Combination* option under *VMware Fusion > Preferences > Hot Keys*, then click on *Fusion Shortcuts* if you need to know what keys release control from the virtual console, or to rebind the shortcut combo.

11.4.4 Virtual Machine Settings

Before students can get started customizing the pfSense operating system for use in their lab environment, there is one more minor edit that needs to be made: Removing the Virtual CD/DVD drive. In the *Virtual Machine Library* window, left click on the pfSense VM entry to highlight it, then right-click on it, and select *Settings*. This window will have two tabs labeled *Hardware* and *Options*. The *Hardware* tab should already be selected by default, but if it is not for some reason, select it. Students should be greeted with a window that looks almost identical to the Hardware window from the *New Virtual Machine Wizard*.

In the *Device* column on the left pane, click on the *CD/DVD (IDE)* entry to highlight it, then click the *Remove* button all the way at the bottom of the left pane. Once finished with this task, Click the *OK* button towards the bottom right corner of the window to close the *Virtual Machine Settings* window.



11-28: Left-click on pfSense in the *Virtual Machine Library* window to highlight it, then right-click and select *Settings*. The *pfSense: Settings* window opens. Click *CD/DVD (IDE)*, then in the *pfSense: CD/DVD (IDE)* submenu, click *Advanced options*, followed by the *Remove CD/DVD Drive* button. As always, a pop-up will appear, asking students to confirm. Click the *Remove* button to proceed. When finished, Close the *pfSense: Settings* window.

11.4.5 pfSense Command-Line and initial interface configuration

In this section, readers will navigate the command-line interface of their pfSense virtual machine to perform essential setup tasks. Once completed, users can navigate to the webConfigurator interface. Begin by powering on the pfSense VM. After a few moments, the boot process completes and students are presented with the pfSense command-line menu. This menu features a series of configuration and troubleshooting options. Begin by selecting option 1 and hitting enter.

11.4.5.1 The Assign Interfaces Wizard

Users are greeted by the *Assign Interfaces* wizard. This wizard is used to map our virtual machine's network interfaces (*Adapter 1*, *Adapter 2*, and *Adapter 3*) to their pfSense aliases – *WAN*, *LAN*, or *OPT1*. Unfortunately, the operating system itself also has unique names for each of these interfaces, adding another layer of complexity and confusion when trying to perform this task.

pfSense itself is based on the FreeBSD operating system, and BSD has its own methods for assigning physical (or virtual, in our case) network interfaces an interface name. For example, BSD assigned the network adapters of my virtual machine the interface names *em0*, *em1*, and *em2*. Every network adapter – integrated or not, virtual or physical, wired or wireless – all have a MAC address to uniquely identify them on a local network. We're going to take advantage of that to know for certain which of the three interfaces *em0* through *em2* map to *Network Adapters 1* through *3*, and how they should be assigned as the *WAN*, *LAN* and *OPT1* aliases. Students were highly advised to record the contents of the *MAC Address* input box of *Network Adapters 1*, *2* and *3* to assist in this task.

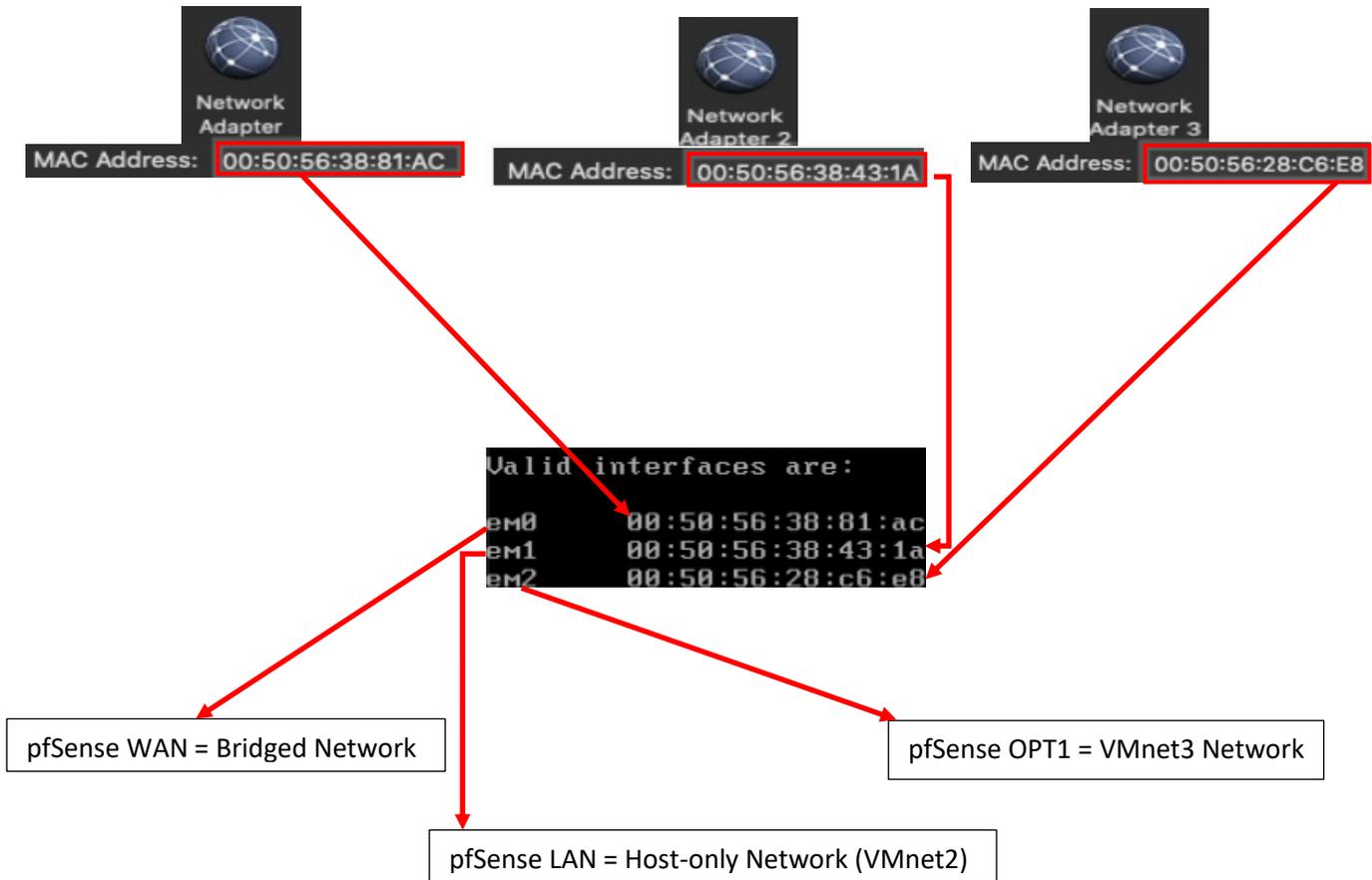
A quick way for readers to determine the interface names for their pfSense installation is through the wizard itself. Upon selecting option 1, a section of text labeled *Valid interfaces are* appears, followed by a series of lines. ***Students should have 3 of these lines in total.*** These lines provide the interface names, MAC addresses, current operational status, and type of hardware BSD identifies the network interface as (The drivers BSD loaded) for each network interface pfSense was able to detect. Here is an example:



```
Valid interfaces are:
1  em0  00:50:56:38:81:ac
2  em1  00:50:56:38:43:1a
   em2  00:50:56:28:c6:e8
```

11-29: A portion of the *Assign Interfaces* wizard. Pay attention to the interface names (1) and the MAC addresses for those interface names (2). This information is needed to determine which virtual network segment they are connected to. This in turn allows students to assign the *WAN*, *LAN* and *OPT1* interfaces correctly.

Compare the MAC addresses displayed, to the MAC addresses recorded earlier, and use that information to complete the rest of the Assign Interfaces wizard. A diagram (fig. 11-30) is provided below to help students understand how to correctly perform this mapping process.



11-30: Here we have the network configuration for my pfSense VM, and the output from the valid interfaces table from the *Assign Interfaces* wizard. *Adapter 1* has the MAC Address 00:50:56:38:81:AC. Looking at the valid interfaces table, em0 has the same MAC address. This means that em0 maps to *Network Adapter 1*, connected to the bridged network. em0 should be assigned as the *WAN* interface. *Adapter 2*'s MAC address matches the MAC address for em1. This means em1 maps to *Network Adapter 2*, connected to the host-only network (VMnet2) – our management network. This means em1 should be assigned the *LAN* interface. Finally, *Network Adapter 3* matches the MAC address for em2. This means em2 maps to the *VMnet3* – IPS 1. This means that em2 should be assigned the *OPT1* interface.

The remainder of this section will aim to guide students through the various questions the wizard will ask (in *italicized* font), and the answers I provided (in *bold* font) based on my lab network and adapter to MAC address mappings. **Students should be aware that this is by and far the most important configuration task for pfSense.** Making sure that the VirtualBox adapters map to the correct pfSense aliases and network segments is absolutely vital to the lab environment working correctly.

Should VLANs be set up now [y|n]? n

*Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): **em0***

*Enter the LAN interface name or 'a' for autodetection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): **em1***

*Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): **em2***

The interfaces will be assigned as follows:

WAN -> em0

LAN -> em1

OPT1 -> em2

*Do you want to proceed [y|n]? **y***

After answering these questions, pfSense will loop back to the main menu.

```

pfSense 2.4.5-RELEASE amd64 Tue Mar 24 15:25:50 EDT 2020
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 5a25c089ce30ec1b5b9e

*** Welcome to pfSense 2.4.5-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.0.39/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:
/64
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 1

Valid interfaces are:

em0      00:50:56:38:81:ac   (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em1      00:50:56:38:43:1a (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em2      00:50:56:28:c6:e8 (down) Intel(R) PRO/1000 Legacy Network Connection 1.

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.
Should VLANs be set up now [yin]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

The interfaces will be assigned as follows:

WAN -> em0
LAN -> em1
OPT1 -> em2

Do you want to proceed [yin]? y

```

11-31: A selection of screen captures from the Assign Interfaces wizard, stitched together to show the questions the wizard asks, and the responses based on network adapter mappings in *fig. 11-30*.

11.4.5.2 Setting IP Addresses for WAN, LAN, and OPT1

The next task we will need to perform on the pfSense command-line is assigning IP addresses to the WAN, LAN, and OPT1 interfaces using the *Set interface(s) IP address* wizard. Most students will have their host system connected to a home or enterprise network where DHCP is available, and just about anything that requests an IP address lease will get one with no problems. That *should* include the pfSense WAN interface bridged to that network. This means the WAN interface should already have an IP address, subnet mask, default gateway (and usually, DNS servers to forward DNS requests to) automatically provided (if this is not the case, see the sidebar discussion, *Help! The WAN Interface has no IP Address*, for some troubleshooting pointers). That means we should only have to run through the Set interface(s) IP address wizard twice – once for the LAN interface, and once for the OPT1 interface. Select option 2 from the pfSense menu to get started.

Similar to the previous section (11.4.5.1), the remainder of this section is going to consist of the questions the *Set interface(s) IP address* wizard will ask students (*italicized*), and the correct answers for the LAN and OPT1 interfaces (in **bold**), followed by an illustration depicting the same questions and answers.

LAN interface:

Available interfaces:

- 1 – WAN ([interface name] – [dhcp/dhcp6/static address configuration])
- 2 – LAN ([interface name] – static)
- 3 – OPT1 ([interface name])

Enter the number of the interface you wish to configure: **2**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.1.1**

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense

e.g. 255.255.255.0 = 24

255.255.0.0 = 16

255.0.0.0 = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.1.10**
Enter the end address of the IPv4 client address range: **172.16.1.254**
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24

You can now access the webConfigurator by opening the following URL in your web browser:

<https://172.16.1.1>

Press <ENTER> to continue. <ENTER>

```
Enter the number of the interface you wish to configure: 2
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 172.16.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.1.10
Enter the end address of the IPv4 client address range: 172.16.1.254
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
      https://172.16.1.1/

Press <ENTER> to continue.
```

11-32: Screen captures from the *Set interface(s) IP address wizard*, stitched together to show the questions the wizard asks, and the responses for the LAN interface based on network adapter mappings in *fig. 11-30*.

Here is an abridged set of *questions* and **answers** for the *OPT1* interface:

OPT1 interface (abridged):

Available interfaces:

- 1 – WAN ([*interface name*] – [*dhcp/dhcp6/static address configuration*])
- 2 – LAN ([*interface name*] – *static*)
- 3 – OPT1 ([*interface name*])

Enter the number of the interface you wish to configure: **3**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.2.1**

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**

Enter the start address of the IPv4 client address range: **172.16.2.10**

Enter the end address of the IPv4 client address range: **172.16.2.254**

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

```

Enter the number of the interface you wish to configure: 3
Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 172.16.2.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new OPT1 IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new OPT1 IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on OPT1? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.2.10
Enter the end address of the IPv4 client address range: 172.16.2.254

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to OPT1...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 OPT1 address has been set to 172.16.2.1/24
Press <ENTER> to continue.

```

11-33: Screen captures from the *Set interface(s) IP address* wizard, stitched together to show the questions the wizard asks, and the responses for the *OPT1* interface based on network adapter mappings in *fig. 11-30*.

After running the wizard again for the *OPT1* interface, students should have an IP address for the *WAN*, *LAN* and *OPT1* interfaces. Additionally, DHCP ranges should be assigned for the *LAN* and *OPT1* interfaces. We're just about ready to move to the webConfigurator, but before doing so, lets run some network connectivity tests first.

```

WAN (wan)      -> em0          -> v4/DHCP4: 10.0.0.197/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:fe23:b366
/64
LAN (lan)     -> em1          -> v4: 172.16.1.1/24
OPT1 (opt1)   -> em2          -> v4: 172.16.2.1/24

```

11-34: The interface information portion of the pfSense command-line menu should look something like this. Looking good is one thing, now let's see if it actually works.

What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?

Unfortunately, I have no way of knowing what network ranges students use at home, so it's entirely possible your physical network may already be using one of the ranges I'm asking you to configure for your lab environment (e.g., 172.16.1.0/24, or 172.16.2.0/24). **To avoid network conflicts on your home network, maybe try these alternate configurations for the *Set interface(s) IP address wizard*:**

Alternate LAN configuration:

LAN interface IP address: 172.16.11.1
Subnet mask bit count: 24
DHCP start address: 172.16.11.10
DHCP end address: 172.16.11.254

Alternate OPT1 configuration:

OPT1 interface IP address: 172.16.12.1
Subnet mask bit count: 24
DHCP start address: 172.16.12.10
DHCP end address: 172.16.12.254

If your lab network is connected to a school or enterprise network using the entire 172.16.0.0/12 allocation, things may be a little more complicated. It may be best to use one of the other RFC1918 network allocations instead, such as 192.168.0.0/16, or 10.0.0.0/8. Why? Enterprise networking can become complicated, either due to growth over time, legacy configurations, or work-arounds to problems accrued over time. You don't want to troubleshoot network problems on your host system, nor do you want the IT ops team coming to your desk over a network outage that could've been avoided. **Here are some alternate configurations for the *Set interface(s) IP address wizard* if you need to avoid using 172.16.0.0/12 entirely:**

Alternate LAN configuration 1:

LAN interface IP address: 10.0.11.1
Subnet mask bit count: 24
DHCP start address: 10.0.11.10
DHCP end address: 10.0.11.254

Alternate OPT1 configuration 1:

LAN interface IP address: 10.0.12.1
Subnet mask bit count: 24
DHCP start address: 10.0.12.10
DHCP end address: 10.0.12.254

Alternate LAN configuration 2:

LAN interface IP address: 192.168.11.1
Subnet mask bit count: 24
DHCP start address: 192.168.11.10
DHCP end address: 192.168.11.254

Alternate OPT1 configuration 2:

LAN interface IP address: 192.168.12.1
Subnet mask bit count: 24
DHCP start address: 192.168.12.10
DHCP end address: 192.168.12.254

Substituting Instructions for Your Chosen Network Ranges

Keep in mind you don't have to use the alternate configurations recommended above. If students have some experience with networking and subnetting, they're welcome to use any network range that suits them. These are just some suggestions to help those who are not quite as experienced, and want to avoid network conflicts.

As a final reminder, **the remaining sections, chapters, and configuration steps will all assume that readers are using 172.16.1.0/24 for the LAN network and 172.16.2.0/24 for the OPT1 network.** This means you will have to mentally substitute steps and commands for the network range you are using instead.

For example, the lab network diagram in chapter 6 has the Kali VM on the IPS 1 (OPT1) network, with an IP address of 172.16.2.2. If you are using an alternate network configuration for the OPT1 network, say 192.168.12.0/24, then the Kali VM's IP address should be 192.168.12.2. If I say "*run the command ssh username@172.16.2.2 to connect to the kali VM*", you'll have to mentally substitute that with `ssh username@192.168.12.2` instead. As another example, firewall rules denying access to or from 172.16.2.3 (Metasploitable2) should be created for 192.168.12.3 instead. Keep this in mind as you continue to build your lab network!

Help! The WAN Interface has no IP Address

If the WAN interface of your pfSense VM has no IP address, consider some of the following to help with troubleshooting:

-No DHCP – It's pretty rare, but perhaps the WAN interface is bridged to a network without DHCP. This just means that you'll have to run the *Set interface(s) IP address* wizard to manually configure the WAN interface IP address, subnet mask, and default gateway. I've already listed the questions the wizard asks, and provided the answers for the LAN and OPT1 interfaces, but since I have absolutely no idea what IP address and subnet mask is assigned to your local physical network, I cannot tell you what you need to enter for the wizard.

If you don't know either, ask a network administrator or whoever is responsible for your network to assist you. Note that if required to manually configure these settings here, practically all of the tasks that require DNS to be configured (e.g., network connectivity tests, and checking for updates on your VMs) will not work until DNS server addresses are configured. This can be done via the webConfigurator, and will be covered shortly.

-Bridged to the wrong host adapter – Another possibility is perhaps the WAN interface virtual adapter may be bridged to the wrong physical network adapter on your host system. If you suspect this is the case, check out [section 11.2](#) (pp. 314-316) for information on how to access the *Virtual Network Editor*, and edit which network interface(s) the Bridged network segment connects to. Then check out [section 11.4.5.1](#) (pp. 337-340) to make absolutely sure that the pfSense WAN interface was mapped to the correct VirtualBox network interface.

-NAC Interference – If you are network security enthusiast at home or connected to an enterprise network, NAC (network access control) may be preventing the WAN interface from obtaining an IP address. Back in chapter 4, [section 4.1.2, NAT Networking \(and Port Forwarding\)](#) (pp. 46-47), readers learned how network address translation works, and how in situations like this, you may be forced to use VMware Fusion's NAT network options to work around network security. If you suspect the WAN interface is being blocked, you can try editing the pfSense VM's settings for the adapter attached to the bridged network, and instead attach it to the *Share with my Mac* network. Open the *pfSense: Settings* menu, select *Network Adapter*, select *Share with my Mac radio button* under the *Internet Sharing* section. Close the settings menu, and reboot the pfSense VM (*Virtual Machine > Restart*). It should have an IP address from the NAT network. Continue below to the next section as normal to test your network connectivity.

Note: If this doesn't work, or attempting to subvert network access controls would otherwise get students in trouble, consider talking to your network/systems administrator and seeing if you can get DHCP allocation and/or necessary exceptions put into place. **Don't violate acceptable use policies, and don't break the law.**

11.4.6 Testing Internet Connectivity using Shell commands

Select option 8, labeled *Shell* in the pfSense menu. Doing so will open up a command-line (bash) shell. Run these 3 commands, and observe their output:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

Here is output from these 3 commands:

```
Enter an option: 8

[2.4.5-RELEASE][root@pfSense.localdomain]/root: ping -c 4 www.google.com
PING www.google.com (172.217.6.100): 56 data bytes
64 bytes from 172.217.6.100: icmp_seq=0 ttl=54 time=24.496 ms
64 bytes from 172.217.6.100: icmp_seq=1 ttl=54 time=22.714 ms
64 bytes from 172.217.6.100: icmp_seq=2 ttl=54 time=21.638 ms
64 bytes from 172.217.6.100: icmp_seq=3 ttl=54 time=19.490 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 19.490/22.084/24.496/1.813 ms
[2.4.5-RELEASE][root@pfSense.localdomain]/root: nslookup www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.6.100
Name:   www.google.com
Address: 2607:f8b0:4009:812::2004

[2.4.5-RELEASE][root@pfSense.localdomain]/root: curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Mon, 01 Jun 2020 02:53:41 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Mon, 01 Jun 2020 02:53:41 GMT
cache-control: private
set-cookie: 1P_JAR=2020-06-01-02; expires=Wed, 01-Jul-2020 02:53:41 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=DdNU16afHrYu25Utm83temwvvrSe6a4UyA3YHz_JKLFzBAv7xrWi8HjSn2-x1PNmxh3EutjAoFBh15hNpxrU72.jpzLLQU0JHJxaOMh5mFyntk5Gae7KUMe2-d1g8I1KloIb7HzOBP_BB4b0sb4lt0Tv1zwOdriVE8ndqfygcrN04; expires=Tue, 01-Dec-2020 02:53:41 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

11-35: The output from the `ping -c 4`, `nslookup`, and `curl -I` commands. All three of these commands completed successfully. Pay close attention to the marked sections above. Note that the IP addresses returned for `nslookup` (In the fields labeled *Address*) may vary based on region.

In a nutshell, these three commands are being used to test various forms of internet connectivity for our VM. `ping -c 4 www.google.com` tells pfSense to send 4 (and only 4) ICMP packets to a specific destination, requesting that the destination respond with its own ICMP packets if it has been reached. `nslookup www.google.com` asks our pfSense virtual machine's configured DNS servers to translate a domain name to an IP address for us. Finally, `curl -I https://www.google.com` is being used to test HTTPS connectivity to the internet. The `-I` option tells the command to only return the HTTP Server headers from our request. All we're really interested in is the line of text: *HTTP/2 200*. This is a thumbs up from Google's webserver confirming that they got our HTTP request with no problems.

Students already familiar with DNS basics may have noticed that we are already trying to ping a domain name (`www.google.com`) with our `ping` command. This means, that in order to actually ping the correct destination, our virtual machine will need to make a DNS request to find the IP address of `www.google.com`. That makes the `nslookup` test redundant, right? Well, yes and no. Later in this chapter, as new virtual machines get created, readers will be advised to perform connectivity tests on those VMs as well. However, the pfSense firewall policy is going to be very strict, so ICMP packets outbound from our lab network will be blocked. Due to how DNS works, the `nslookup` check can still be used to make sure VMs can resolve domain names, and the `curl` connectivity test will be more than sufficient to confirm whether or not lab virtual machines have the internet access they require. After performing these commands and confirming internet connectivity, type `exit` to leave the shell.

My connectivity commands failed! Now what?

If students got anything other than output similar to *fig. 11-35* (e.g., request timeouts and/or packet loss for `ping`, timeouts for `nslookup`, no response for `curl -I`), then there are connectivity issues to be sure. Troubleshooting network connectivity is an extremely complex topic. I can't give you a definitive guide for finding the root of your problem, but I can tell you to start with the basics and work your way up – sometimes the cause of your network problems are settings or hardware that was taken for granted.

Checking physical cabling, link lights and physical connectivity to network devices always comes first. As an extension to that, check out the sidebar in section 11.4.5.2 (*Help! The WAN interface has no IP address*) for some additional clues. The VM may be bridged to the wrong physical adapter. Some form of network security (e.g., a network firewall) may be preventing your VM from connecting to the internet. Try connecting the *Network Adapter* to a NAT (*Share with my Mac*) network instead. The incorrect network adapter may have been chosen to be the WAN interface. Consider re-running the *Assign Interfaces* wizard again, and compare the MAC addresses from the wizard to the MAC addresses of the network adapter in the pfSense VM's Settings menu. See *fig. 11-30* for guidance on confirming that interfaces have been mapped correctly.

If students were required to run the *Set interface(s) IP address* wizard for the WAN interface (No DHCP), or your local network's DHCP server doesn't assign DNS servers automatically, your troubleshooting commands will fail because pfSense has no way of resolving domain names. We will be covering how to manually configure a primary and/or secondary DNS server for pfSense via the webConfigurator shortly.

If your host system is connected to a physical network already using 172.16.1.0/24 or 172.16.2.0/24, you may be experiencing network conflicts, routing loops, or other weird behavior. Assign different IP addresses and ranges to the LAN and OPT1 networks to avoid network conflicts. See the sidebar discussion in 11.4.5.2 labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?*

Last but not least, check and double check that you entered the commands correctly. Typos matter on the command-line, and BSD will not hold your hand if the command is entered incorrectly. If all else fails, don't be afraid to ask others for guidance.

11.4.7 Finish setting up pfSense

Navigate to [chapter 14, *pfSense Firewall Policy and Network Services*](#), starting on *p.* 664 and follow the chapter guidance. Once completed, readers will be directed back here to complete their lab environment.

11.5 Create the Remaining Virtual Machines

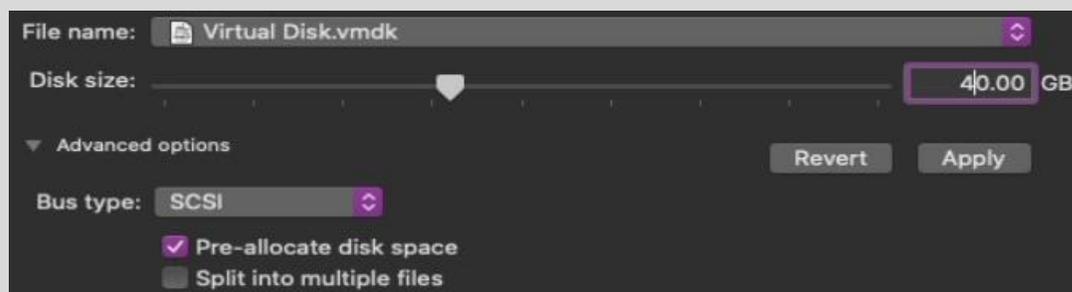
Welcome back! Now that the pfSense VM is fully functional, it's time to start working on the remaining lab VMs. In this section, users will create three of the four remaining virtual machines via the *Create Virtual Machine* wizard, then adjust the *Settings* of each virtual machine. After the SIEM, IPS and Kali VMs are created and configured, readers will be guided through the operating system installation, and initial setup process for all three VMs. The Metasploitable 2 VM is a unique case, and will be covered separately.

11.5.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali

Run the *New Virtual Machine Wizard* three times, with the *Create a custom virtual machine* button selected, and the settings listed in the table below. Assume the default for any settings not mentioned in the table below. Refer back to [section 11.4.1](#) (pp. 318-321) for guidance on how to access and progress through the wizard as needed.

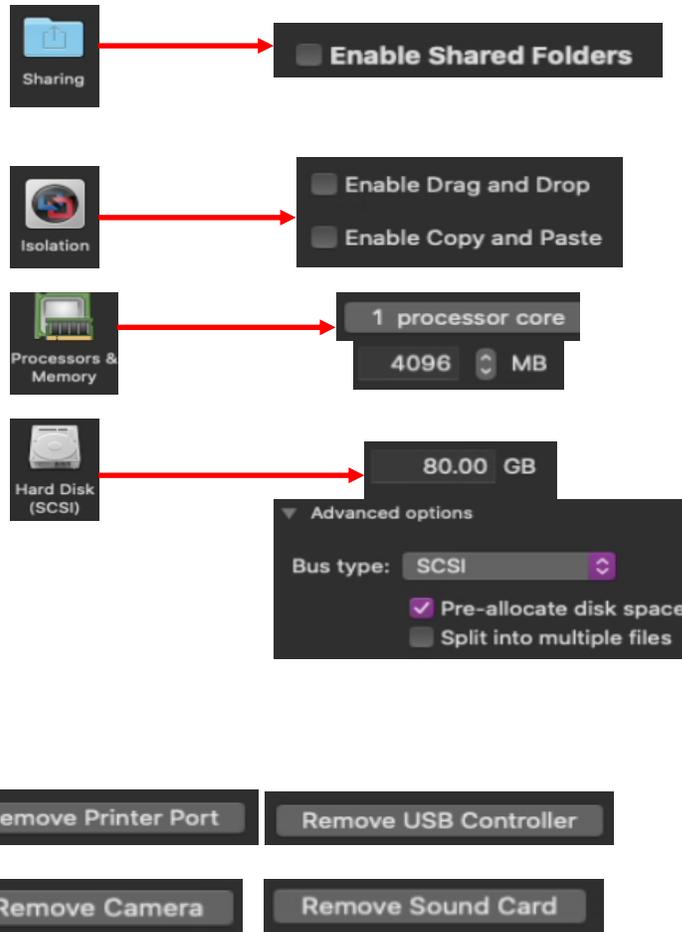
Close Quarters Storage in Tight Spaces

In section 11.4.2, in the sidebar conversation, [The Basics of Close Quarters Storage](#) (p. 323), I warned that most MacBooks are lacking disk space. For example, if your MacBook has a 250GB SSD (like mine), it's really easy to lose about 50-60GB from the operating system, and a few core applications. That's before we even worry about storing virtual machine VMDK files. If you are looking for a way to save disk space, remember in Chapter 6, that I made the recommendation to cut the disk space allocations for the SIEM, IPS, and Kali VMs in half – from 80GB down to 40GB if disk space is tight. Unfortunately, this may mean having to monitoring virtual machines more closely to make sure their disks don't fill and lead to stability and service problems, but this is the price to pay. Additionally, while we haven't covered this subject just yet, **VM snapshots take space as well. How much space depends on the size of the VMDK, and the amount of data written to the VMDK. That disk space you saved may be the difference between being able to snapshot your virtual machines, and scrambling for disk space.**



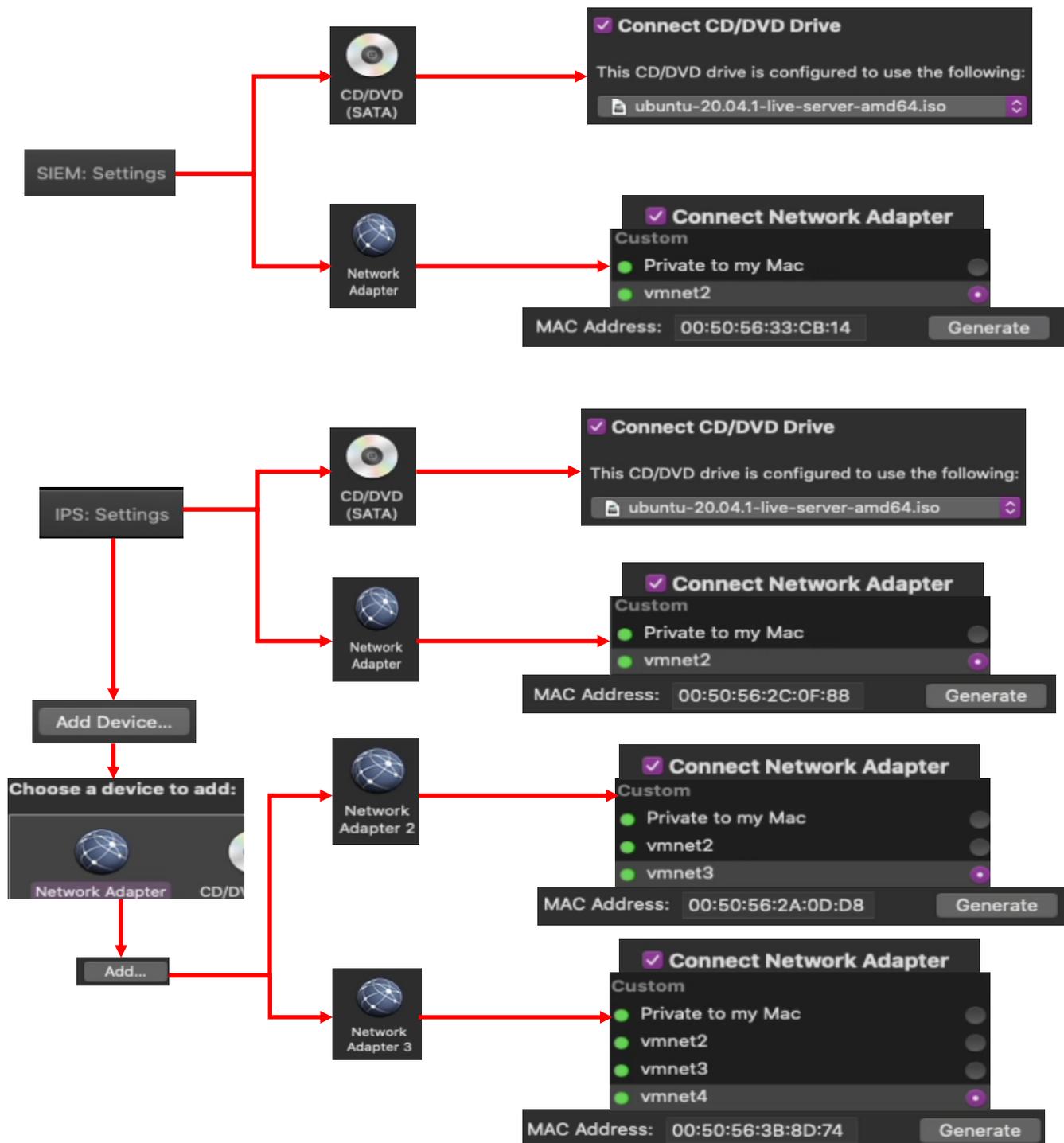
11-36: Remember that if you are severely lacking in storage, that the disk allocations (e.g., *Disk Size*) for the SIEM, IPS, and Kali VMs can be reduced from 80GB, down to 40GB. This may be enough of a size reduction to allow for pre-allocation of disk space, and proper snapshots.

Choose Operating System:	Linux Ubuntu 64-bit	Linux Ubuntu 64-bit	Linux Debian 10.x 64-bit
Choose Firmware:	Legacy BIOS	Legacy BIOS	Legacy BIOS
Choose a Virtual Disk:	Create a new virtual disk	Create a new virtual disk	Create a new virtual disk
Save As:	SIEM	IPS	Kali
Customize Settings:			
-Processors & Memory:	4GB (4096MB) 1 processor core	4GB (4096MB) 1 processor core	4GB (4096MB) 1 processor core
Sharing	Verify <i>Enable Shared Folders</i> checkbox is unchecked	Verify <i>Enable Shared Folders</i> checkbox is unchecked	Verify <i>Enable Shared Folders</i> checkbox is unchecked
-Hard Disk (SCSI)	80.00GB	80.00GB	80.00GB
-Isolation	Uncheck <i>Enable Drag and Drop</i> <i>Enable Copy and Paste</i>	Uncheck <i>Enable Drag and Drop</i> <i>Enable Copy and Paste</i>	Uncheck <i>Enable Drag and Drop</i> <i>Enable Copy and Paste</i>
-CD/DVD:	In the <i>This CD/DVD drive is configure to use the following</i> drop-down menu, select <i>Choose a disc or disc image</i>	In the <i>This CD/DVD drive is configure to use the following</i> drop-down menu, select <i>Choose a disc or disc image</i>	In the <i>This CD/DVD drive is configure to use the following</i> drop-down menu, select <i>Choose a disc or disc image</i>
	Locate the Ubuntu Server ISO, and select it.	Locate the Ubuntu Server ISO, and select it.	Locate the Kali Linux ISO, and select it.
	Ensure <i>Connect CD/DVD Drive</i> checkbox is checked	Ensure <i>Connect CD/DVD Drive</i> checkbox is checked	Ensure <i>Connect CD/DVD Drive</i> checkbox is checked
-Number of Network Adapters:	1	3 (Click <i>Add Device</i> button, select <i>Network Adapter</i> twice)	1
-Network Adapter Settings:	Network Adapter: Select <i>vmnet2</i> radio button. Generate and document a new MAC address. (<i>Advanced options</i>)	Network Adapter: Select <i>vmnet2</i> radio button. Network Adapter 2: Select <i>vmnet3</i> radio button. Network Adapter 3: Select <i>vmnet4</i> radio button. Generate and document a new MAC address for each adapter. (<i>Advanced options</i>)	Network Adapter: Select <i>vmnet3</i> radio button. Generate and document a new MAC address. (<i>Advanced options</i>)
-Remove the following virtual hardware:	USB Controller Sound Card Printer Camera	USB Controller Sound Card Printer Camera	USB Controller Sound Card Printer Camera

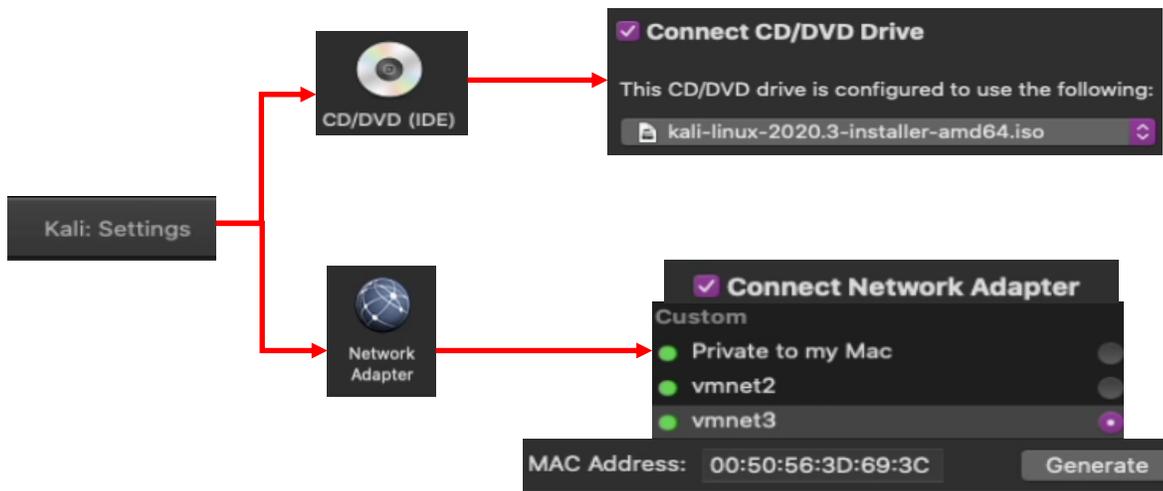


11-37: Using the table above, create three new virtual machines, using the new virtual machine wizard. All of the newly created virtual machines should have the following settings in common:

- Sharing*: ensure the *Enable Shared Folders* checkbox is unchecked
- Isolation*: uncheck the *Enable Drag and Drop*, and *Enable Copy and Paste* options
- Processors & Memory*: 1 processor core, 4096MB of RAM
- Hard Disk (SCSI)*: set the *Disk Size* to 80.00GB
- Remove the following hardware: Printer, USB Controller, Camera, and Sound Card



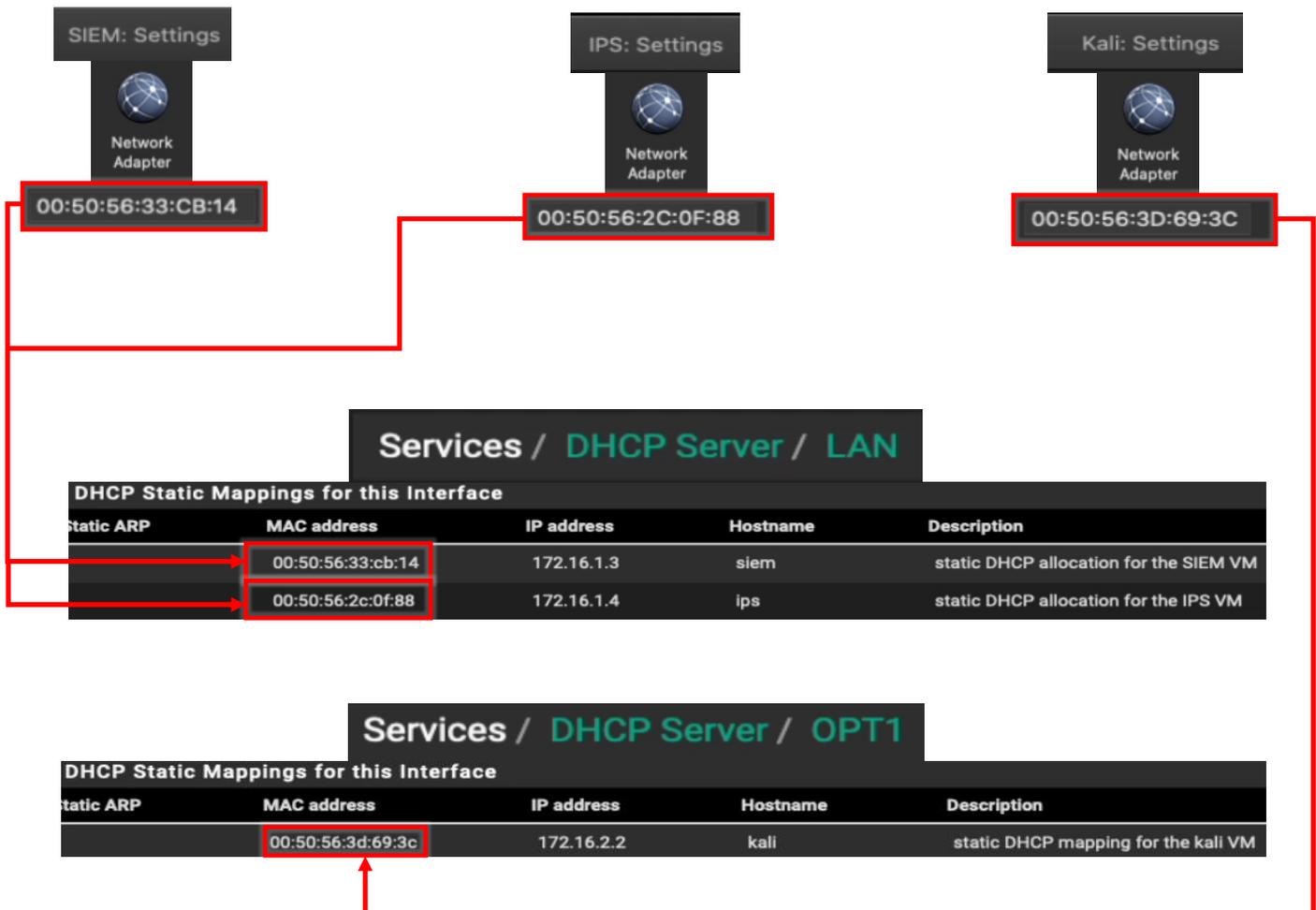
11-38: Depicted in the illustration above are virtual machine settings for the SIEM and IPS VMs. Ensure that the *CD/DVD (SATA)* device is configured to use the Ubuntu Server 20.04 ISO and that the *Connect CD/DVD Drive* checkbox is checked for both the SIEM and IPS virtual machines. Connect the SIEM VM's *Network Adapter* to *vmnet2*, ensure the *Connect Network Adapter* checkbox is checked, then *Generate* and record the MAC address. For the IPS VM, students will need to add two additional network adapters through the *Add Device* button. Attach *Network Adapter 1* to *vmnet2*, *Network Adapter 2* to *vmnet3*, and *Network Adapter 3* to *vmnet4*. *Generate* and document the MAC address, and ensure the *Connect Network Adapter* checkbox is checked for all three adapters.



11-39: Last but not least, here are the virtual machine settings specific to the kali VM. Students will need to configure the CD/DVD (IDE) device to use the Kali Linux ISO, and ensure the Connect CD/DVD Drive checkbox is checked. The Kali VM's *Network Adapter* should be attached to vmnet3. As with the other virtual machines, ensure the *Connect Network Adapter* checkbox is checked, then *Generate* and record the MAC address for the interface.

11.5.2 Creating Static DHCP Allocations for the SIEM, IPS and Kali VMs

With all three virtual machines created, students will log on to the pfSense WebConfigurator and configure three static DHCP mappings. Students may refer to Chapter 14, [section 14.3.4.1](#) (pp. 690-692) for a refresher on creating static DHCP mappings on pfSense. The SIEM VM should be statically assigned the IP address 172.16.1.3, and the vmnet2 interface of the IPS VM should be assigned 172.16.1.4. Both of these allocations should be configured on the *LAN* interface of the IPS VM. Meanwhile, the Kali VM should be assigned the IP address 172.16.2.2 on the *OPT1* interface.



11-40: Log in to the pfSense WebConfigurator, and create a static DHCP mapping for the SIEM VM and the vmnet2 adapter of the IPS VM on the LAN interface, and a static DHCP mapping for the kali VM on the OPT1 interface.

11.5.3 Operating System Installation

In this section, students will learn how to install the operating system for the SIEM, IPS, and Kali virtual machines. Both the SIEM and IPS VMs will have Ubuntu Server installed as their operating system, while the Kali VM will have the latest version of the Kali Linux distribution installed. The installation instructions will differ for each virtual machine, so please pay attention.

As a general reminder, please make sure that the pfSense VM is running, and that you have completed chapter 14 to ensure pfSense is ready to support the rest of the lab environment. Without the pfSense VM, none of the virtual machines will have internet access. That may result in the operating system installers failing in different ways. To confirm that pfSense is running, students can check tiny icon next to the name of the VM on the *Virtual Machine Library* window. If the VM is running, A small green arrow will appear over a small white box.



11-41: The little green arrow head icon is a quick indicator that a VM is powered on and running. The other indicator is the virtual console display in the *Virtual Machine Library* window.

11.5.3.1 Installing Ubuntu on the SIEM VM

To get started, on the *Virtual Machine Library* window, click the SIEM entry to highlight it, then right-click on the entry, and select *Start Up*. The virtual machine will begin booting off the Ubuntu Server ISO. The first screen students see will ask to confirm the language they wish to use. The default language should be *English*, so hit the enter key on your keyboard to continue.

Depending on when students downloaded their copy of the Ubuntu Server ISO, and how frequently the ISO is updated, a screen may appear titled *Installer update available*. This screen provides users with the option to download the latest version of the Ubuntu installation wizard, called Subiquity. Use the arrow keys on your keyboard to highlight *Update to the new installer*, then hit enter.

Note: If for some reason downloading the latest installer fails, there's a good chance that there are network problems with the lab environment elsewhere, and that there is troubleshooting to do. Students are welcome to select the *Continue without updating* option, but keep this in mind if the installer misbehaves or fails later. Check to see if the hypervisor host has internet connectivity, double check the firewall rules on the pfSense virtual machine, network settings, physical cabling, etc.

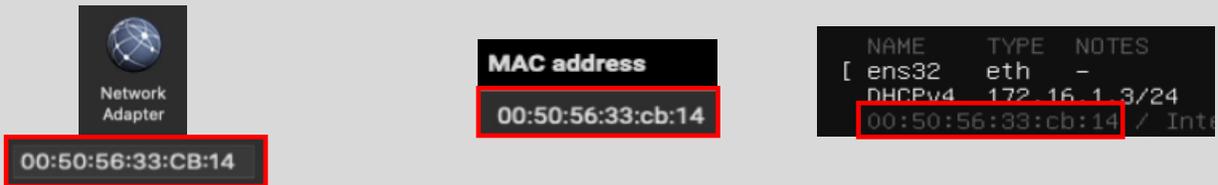
The next screen asks users to confirm their keyboard configuration. The default settings for both the *Layout* and *Variant* settings are *English (US)*. If you are not using a standard US-English keyboard, you may wish to use the arrow keys to highlight the *Identify keyboard* option, then

hit enter. Otherwise, highlight *Done* on the bottom of the screen, and hit enter.

Next up, is the *Network connections* screen. a single network adapter should populate this page. The network adapter (named *ens32* in my case) should automatically be assigned the IP address 172.16.1.3. Below the IP address in light grey text is the MAC address of the network adapter that the Ubuntu installer detected. This should be the same MAC address of the network adapter of the SIEM VM. If the correct IP address was assigned, students can hit the enter key to continue (The *Done* option should be highlighted by default). Otherwise, see the section *What Reservation?* for some troubleshooting tips.

What Reservation?

If for some reason the network adapter was assigned any other IP address other than 172.16.1.3, Refer back to section 11.5.2. Check the *MAC Address* field in the *Network Adapter* menu for the SIEM VM. Compare that MAC address to the MAC address used to create a static DHCP mapping on the *LAN* interface of the pfSense VM. Compare that to the MAC address displayed on the *Network connections* screen of the Ubuntu installer. **They should all be identical.** If there are any errors, correct them and restart the SIEM VM, to restart the Ubuntu installer until the pfSense DHCP assigns the network adapter the correct IP address.



11-42: If the SIEM VM failed to get the correct IP address, check the *Edit settings* menu of the virtual machine – specifically the *MAC Address* field under *Network Adapter 1*. Compare that to the MAC address used to create a static DHCP mapping on the *LAN* interface on the pfSense WebConfigurator. Correct the static DHCP entry as necessary then restart the SIEM VM to restart the ubuntu installer. Confirm that the network adapter was correctly assigned the 172.16.1.3 IP address.

The *Configure proxy* screen appears. Use the up arrow key to highlight the text box labeled *Proxy address* and enter `http://172.16.1.1:3128`. If you recall from Chapter 14, this is the IP address and port for the Squid proxy on the *LAN* interface of the pfSense VM. Use the arrow keys to highlight *Done*, and hit enter to continue.

The next screen, labeled *Configure Ubuntu archive mirror* will appear. This is another one of those situations where students will know whether or not they need to change this setting. Unless the lab environment is in an enterprise network and the network team happens to be operating their own software archive mirror, accept the default setting (in my case, the default mirror address was `http://us.archive.ubuntu.com/ubuntu`). With *Done* highlighted, hit enter to continue.

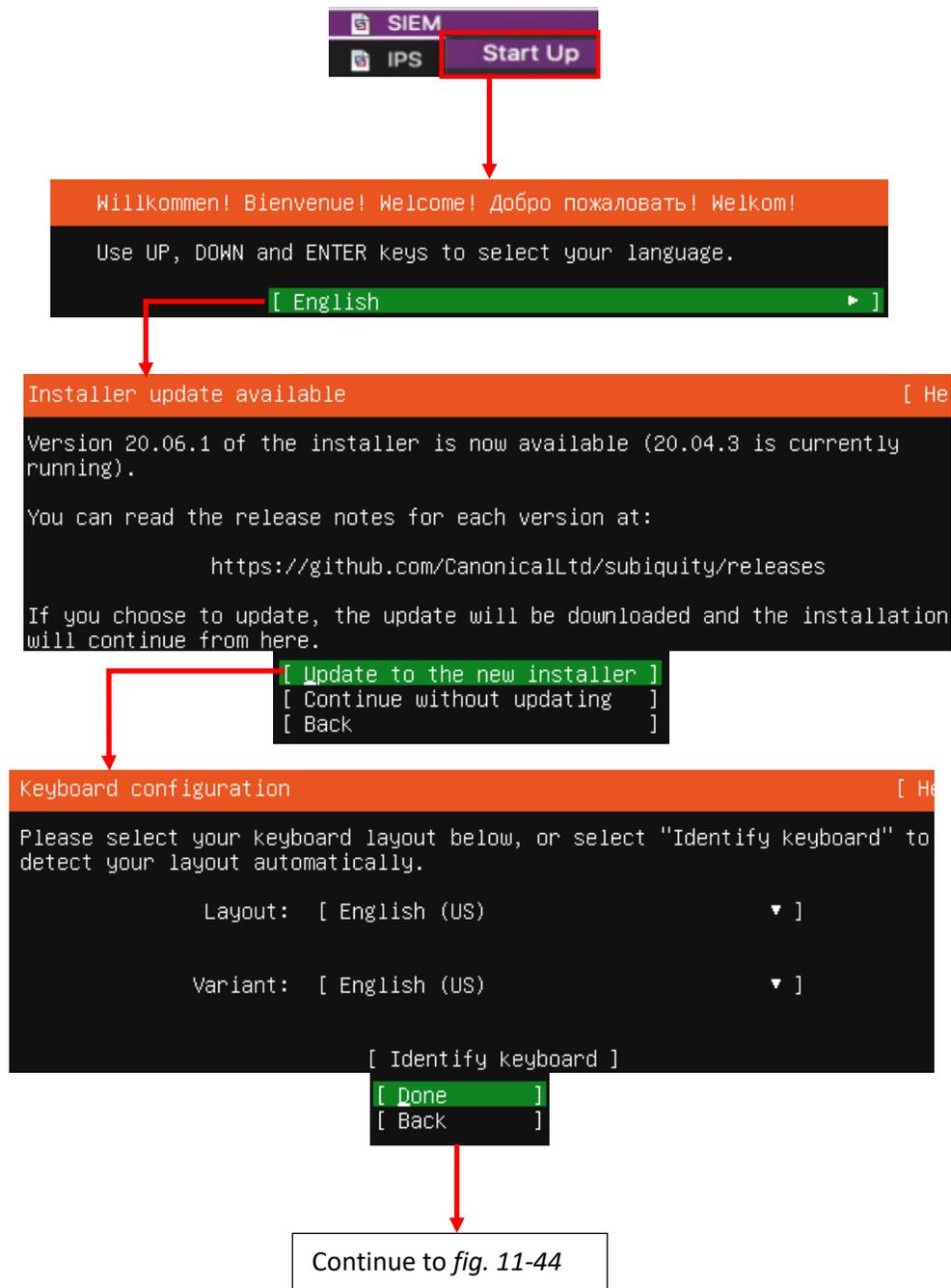
The *Guided storage configuration* screen appears next. Accept the default settings and let the Ubuntu installer format the entire disk. Use the arrow keys to highlight *Done*, and hit enter to continue. The next screen, titled *Storage configuration*, shows you how the installer is going to format the hard drive, and what partitions are going to be where in a large list labeled *FILE SYSTEM SUMMARY*. By default, *Done* should already be highlighted on this screen. If not, use the arrow keys to highlight it and hit enter to continue. A pop-up labeled *Confirm destructive action* appears. This screen informs the user that any data on the disk will be lost as a result of formatting and partitioning the disk. Since there is no data on the virtual hard disk yet, with *Continue* highlighted, hit the enter key to proceed.

Next is the *Profile setup* screen. There are five input boxes on this screen. Ubuntu asks the user for their name, the server's name, a username (that will be used to log in to the server later), the password for that username, followed by an input box asking the user to repeat the password. Students may enter any name, username, or password they would like, but it is recommended to both set the server name to *siem*, as well as to save the username and password combination to a password manager. Once finished, use the arrow keys to highlight *Done*, then hit enter to continue.

The *SSH Setup* screen appears and asks users if they would like to install the OpenSSH server package by default. By default, the prompt should be between two brackets next to the text *Install OpenSSH server*. Hit the spacebar to leave an 'X' between the brackets. Afterwards, use the arrow keys to highlight *Done* and hit enter to advance.

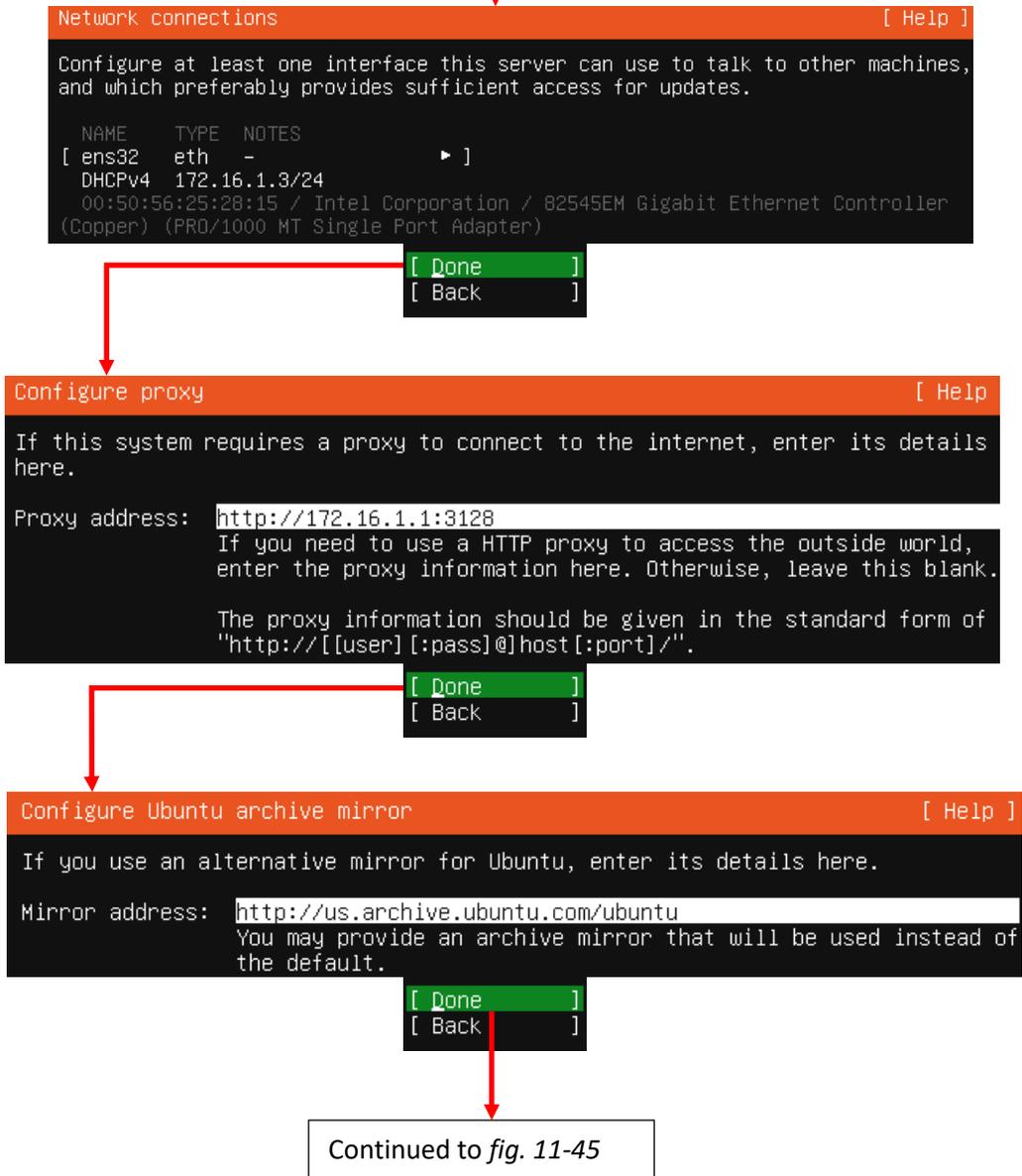
The next screen is labeled *Featured Server Snaps*. The latest versions of Ubuntu use an additional software manager called 'snap' to deliver software packages. Use either the arrow keys or the tab key to highlight *Done* and hit enter – Do not install any snaps, continue the installer.

We reach a new screen labeled *Install complete!* At this point, students have made all of the necessary decisions for the ubuntu installer to proceed, and handle all of the installation tasks at once. Once completed, the installer will grant students the option to reboot the system. However, instead of using the installer's reboot function, click the *Actions* option from the virtual console window, followed by *Power*, then *Power off* to shut down the virtual machine. Afterwards, close the virtual console.



11-43: In the *Virtual Machine Library* window, left click on the SIEM VM entry, then right-click and select *Start Up*. The virtual console should appear automatically. The user then selects the language of the installer, the installer checks for updates for itself, then asks the user to set the keyboard layout and variant language.

Continued from *fig. 11-43*



11-44: The next stages of the Ubuntu Server 20.04 installer. In these screens, students can confirm whether or not the static DHCP mapping for the SIEM VM is working correctly, configure the system to use the Squid proxy service configured on the pfSense VM, and confirm software archive mirror they would like to use.

Continued from *fig. 11-44*

```
Guided storage configuration
Configure a guided storage layout, or create a custom one:
(X) Use an entire disk
    [ /dev/sda local disk 80.000G ▾ ]
[X] Set up this disk as an LVM group
    [ Done ]
    [ Back ]
```

```
Storage configuration [ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT  SIZE  TYPE  DEVICE TYPE
[ /          39.498G new ext4 new LVM logical volume ▶ ]
[ /boot     1.000G new ext4 new partition of local disk ▶ ]

AVAILABLE DEVICES
DEVICE                                TYPE                                SIZE
[ ubuntu-vg (new)                    LVM volume group                  78.996G ▶ ]
free space                            39.498G

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES
DEVICE                                TYPE                                SIZE
[ ubuntu-vg (new)                    LVM volume group                  78.996G ▶ ]
ubuntu-lv  new, to be formatted as ext4, mounted at /  39.498G ▶ ]
[ /dev/sda                            local disk                        80.000G ▶ ]
partition 1 new, bios_grub                                       1.000M ▶ ]
partition 2 new, to be formatted as ext4, mounted at /boot 1.000G ▶ ]
partition 3 new, PV of LVM volume group ubuntu-vg      78.997G ▶ ]
```

```
[ Done ]
[ Reset ]
[ Back ]
```

Continued to *fig. 11-46*

11-45: These screens are used to configure the storage settings for the operating system. Students will be using the default storage settings for the SIEM VM.

Continued from *fig. 11-45*

```
Confirm destructive action

Selecting Continue below will begin the installation process and
result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the
installation has started.

Are you sure you want to continue?

[ No ]
[ Continue ]
```

```
Profile setup [ Help ]

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

Your name: ayy
Your server's name: siem
The name it uses when it talks to other computers.
Pick a username: ayy
Choose a password: *****
Confirm your password: *****

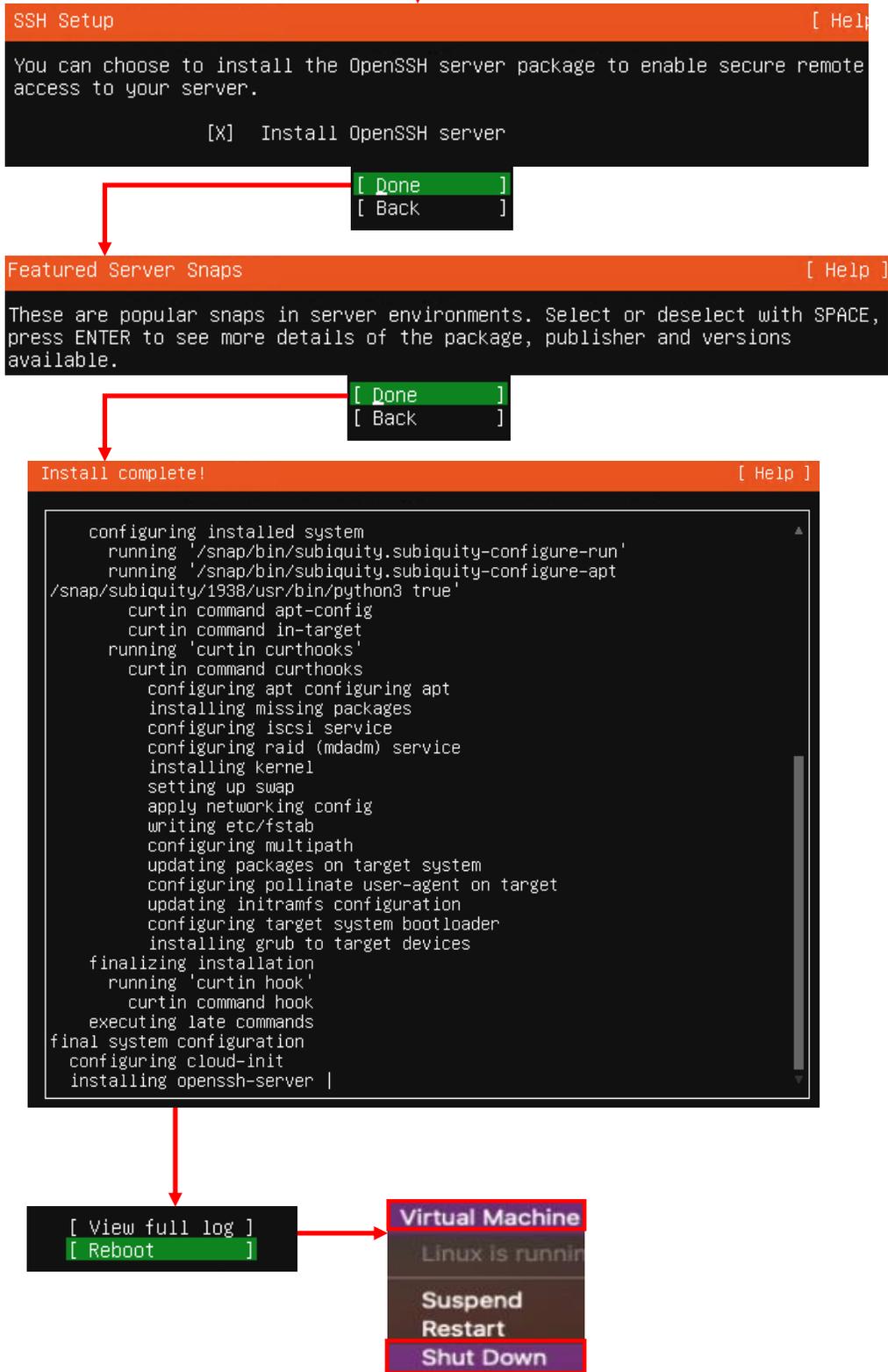
[ Done ]
```

Title:	SIEM VM
Username:	ayy
Password:	*****
URL:	172.16.1.3
<input type="checkbox"/> Expires:	7/21/2020 12:11 PM
<input checked="" type="checkbox"/> Notes:	User account credentials for the SIEM VM.

Continued to *fig. 11-47*

11-46: After confirming the storage configuration settings, users are prompted to name their server, and create a user account. It's recommended to store the username and password for the SIEM VM in a password manager.

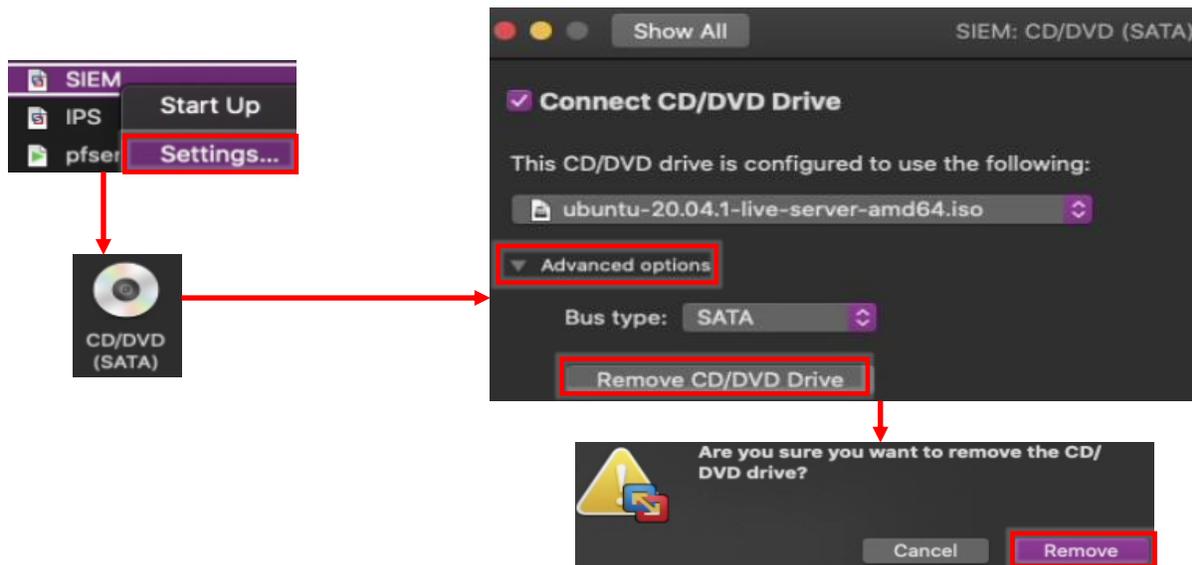
Continued from *fig. 11-46*



11-47: The final stages of the Ubuntu Server 20.04 installer. Select the option to install OpenSSH server, then decline to install any server snaps. Finally, once the installer grants you the option to reboot, select Virtual Machine > Shut Down from the navigation menu to power down the VM.

11.5.3.2 Additional Virtual Machine Settings – SIEM VM

Now that the operating system installation is complete, there is one last configuration setting to adjust on the SIEM VM before we can boot into Ubuntu Linux and perform some diagnostic tasks. Back in [section 11.4.4](#) (p. 336), students learned how to remove the *CD/DVD Drive* from the pfSense VM. Students need to perform that task on the SIEM VM. Open the SIEM VM's *Settings* menu, locate the *CD/DVD (SATA) Device*, select *Advanced options*, then click the *Remove CD/DVD Drive* button. Afterwards click the *OK* button in the lower right corner to confirm this change, and close the SIEM VM's settings menu.



11-48: Left-click on pfSense in the *Virtual Machine Library* window to highlight it, then right-click and select *Settings*. The *SIEM: Settings* window opens. Click *CD/DVD (SATA)*, then in the *CD/DVD (SATA)* submenu, click *Advanced options*, followed by the *Remove CD/DVD Drive* button. As always, a pop-up will appear, asking students to confirm. Click the *Remove* button to proceed. When finished, Close the window.

11.5.3.3 Booting the SIEM VM for the first time

After changing the SIEM VM's settings, start the VM back up and bring up its virtual console. After a moment or two, you will be greeted with login prompt labeled *SIEM login*. Enter the username you configured during the installer, followed by the password to log in.

Some students may not be familiar with command-line applications, and that's okay. This is only a quick login to make sure network connectivity is working. Please type in the following commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The purpose of the `ip` command above is to display all of the network interfaces on the system. We pass this command the `-br` option for brief output, followed by the letter 'a' to indicate we're interested in seeing the IP addresses on our system. Users could replace 'a' with 'address' or 'addr' and the `ip` command would interpret it the same. We're using this command to serve as a secondary confirmation that the SIEM VM was successfully assigned the IP address 172.16.1.3, as displayed in *fig.11-47* below. Students may notice a second interface on the system designated `lo`. This is a "loopback" network interface and can safely be ignored.

The `nslookup` command is to confirm that the SIEM VM is able to resolve hostnames using DNS. The output from the command should be similar to what is presented in *fig. 11-47*. Finally, that brings us to the `curl` command. This command is to confirm connectivity to the internet over port 443, HTTPS. The `-I` option in the command tells `curl` to only return the headers from the web server being contacted. Once again, the output of this command should be fairly similar to what is presented in *fig. 11-47*.

```
Ubuntu 20.04 LTS siem tty1
siem login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ nslookup www.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:806::2004
ayy@siem:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 21 Jul 2020 20:10:38 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 21 Jul 2020 20:10:38 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-21-20; expires=Thu, 20-Aug-2020 20:10:38 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=1AAB5nk21PEgo8rGiFr-9PxEuTIYONxZtMMi-EmACtdRnP1PkB0xoosGu9FjWzbLyW0TGOHKUj6kLonn4Rr-yu-Mic8itYAIIS07X2VkJb1WkOUK30rSk6Fd0ce6_Battd9PQ4YI7-CoRGf381n74m078YmTAAAtz1XJf8-WM; expires=Wed, 20-Jan-2021 20:10:38 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@siem:~$ ip -br a
lo                UNKNOWN      127.0.0.1/8  ::1/128
ens32             UP          172.16.1.3/24 fe80::250:56ff:fe33:cb14/64
```

11-49: After logging in to the SIEM virtual machine, students will need to run a series of network troubleshooting commands. These commands are to confirm that the SIEM VM has the correct IP address configured (`ip -br a`), can resolve hostnames via DNS (`nslookup www.google.com`), and has connectivity over HTTPS (`curl -I https://www.google.com`).

Before logging out of the SIEM virtual machine, there are three more commands to run, but before we can run them, we will need to become the root user. Enter the following command:

```
sudo su -
```

When prompted, enter the password for the user you created. If successful, you will be logged in as the root user on the SIEM virtual machine. The root user, sometimes referred to as the super user, is a special account that has complete authority over the system. Additionally, root has access to special administrative commands that normal users are not allowed to use. As the root user, let's **run those last three commands in this exact order**:

```
apt-get update
apt-get -y dist-upgrade
init 6
```

Ubuntu is based off of the Debian Linux distribution. Because of this, it uses a package manager called apt (in addition to the snap package manager mentioned earlier). The two apt-get commands, apt-get update then apt-get -y dist-upgrade tell Ubuntu to reach out to the software archive mirror and get an updated list of software packages, then if any packages installed on the system need to be updated, updated them immediately. This set of commands also confirms that the Squid proxy server on the pfSense VM is working properly, and proxying all of the HTTP requests from the SIEM VM. The final command, init 6, tells the system to reboot immediately. As an alternative, users can also run the command reboot instead.

```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (524 kB/s)
Reading package lists... Done
root@siem:~# apt-get -y dist-upgrade
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@siem:~# init 6_
```

11-50: the command `sudo su -` allows students to become the root user on the SIEM VM. We then use root's permissions to ensure all of the software packages on the system are up to date (apt-get update, followed by apt-get -y dist-upgrade), then immediately reboot the system (init 6, or optionally reboot). Be aware that the apt-get commands may take a little bit of time to finish, based on the number of updates available and speed of your internet connection.

Help! My apt-get commands are failing!

If you're experiencing problems with the apt-get commands failing to complete, there's a very good chance that the apt package manager is not properly configured to use the SQUID HTTP proxy we installed on the pfSense VM, or that the SQUID proxy service on the pfSense VM may be misconfigured. If students entered the wrong information during the operation system installation (e.g. on the Configure Proxy screen), then the apt package manager will not work properly.

Here are some troubleshooting steps to think about:

On the SIEM VM, run the command:

```
cat /etc/apt/apt.conf.d/90curtin-aptproxy
```

This command will read the contents of the file /etc/apt/apt.conf.d/90curtin-aptproxy and display its contents on the screen. The file should read something like this:

```
Acquire::http::Proxy "http://172.16.1.1:3128";  
Acquire::https::Proxy "http://172.16.1.1:3128";
```

If this file does not exist, or has any content that is in any way different from the lines above, **run the following three commands exactly as displayed, and in this exact order:**

```
sudo su -  
echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-  
aptproxy  
echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >>  
/etc/apt/apt.conf.d/90curtin-aptproxy
```

This series of commands requires root access, so the first thing we do is use `sudo su -` to become the root user. The next two commands delete the current `90-curtin-aptproxy` file if it exists, then overwrites it with the two correct entries that should exist in the file. After running these commands, run `cat /etc/apt/apt.conf.d/90curtin-aptproxy` once more, and confirm that the output matches the correct output listed above. After confirming that the configuration file has been recreated correctly, try running the apt-get commands once more. If they continue to fail, then continue the troubleshooting process. Assuming that the network connectivity check commands were successful (e.g. `nslookup` and `curl`), think about the following:

- Is the SQUID proxy service installed on pfSense?
- Is there a firewall rule on the LAN interface to allow access to the proxy service? (allow traffic to IP address 172.16.1.1 port 3128 TCP from network 172.16.1.0/24)
- Is the option *Resolve DNS IPv4 First* checked on the SQUID proxy service?

These are all configurations covered in chapter 14, and should have already been specified. Double check that they have been configured correctly, then try updating the SIEM VM again.

```

ayy@siem:~$ 1 sudo su -
root@siem:~# 2 echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 3 echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >> /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 4 cat /etc/apt/apt.conf.d/90curtin-aptproxy
Acquire::http::Proxy "http://172.16.1.1:3128";
Acquire::https::Proxy "http://172.16.1.1:3128";
root@siem:~# 5 apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [332 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [8,780 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [5,404 B]
Fetched 826 kB in 1s (972 kB/s)
Reading package lists... Done
root@siem:~# 6 apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@siem:~#

```

11-51: This illustration demonstrates how to fix or modify the `/etc/apt/apt.conf.d/90curtin-aptproxy` file, in the event that student find that there is a problem with the file. First utilize `sudo su -` (1) to become the root user. Then use the two `echo` commands (2, 3) to write the correct configuration data so that `apt` knows how and where to access the squid proxy configured on the pfSense VM. Utilize the `cat` (4) command to confirm that the configuration file is properly configured. Finally, run `apt-get update` (5) and `apt-get -y dist-upgrade` (6) to check for the latest updates and download them.

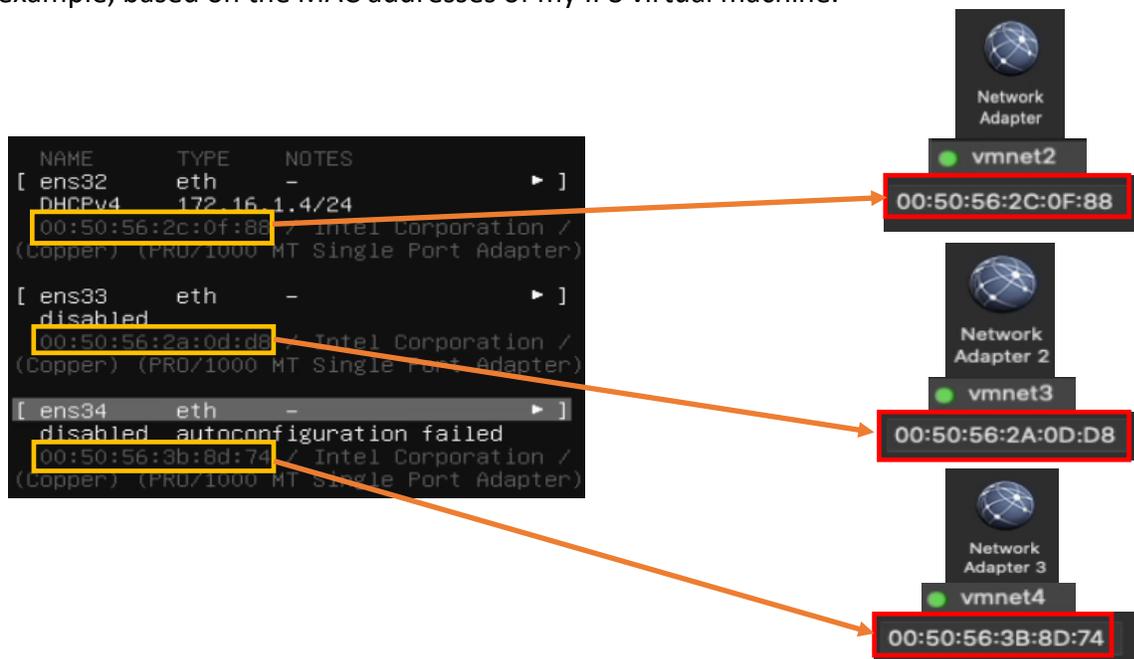
11.5.3.4 Installing Ubuntu on the IPS VM

Now that Ubuntu has been installed on the SIEM VM, network connectivity has been checked, and updates have been applied, next up is the IPS VM. The process for installing Ubuntu Server on the IPS virtual machine is practically identical, the process will be summarized below, with major differences to be aware of explained further in-depth.

- Start the IPS VM, and connect to its virtual console
- Select English as your language (or your preferred language)
- If there are any updates to Subiquity, select the option, *Update to the new installer*
- Select *English (US)* (or your preferred language) as the keyboard *Layout* and *Variant*

The *Network connections* screen will be a little bit different than it was on the SIEM VM, because the IPS virtual machine has three network interfaces. Recall in section 11.4.5.1, comparing and contrasting the MAC addresses of the three network adapters attached to the pfSense VM, and using that information to correctly map the network interfaces to the correct networks.

Students will need to perform a similar exercise for the IPS virtual machine on the *Network connections* screen. In light grey text underneath the name of each network interface is the MAC address for that interface. Cross-reference the MAC address and interface name on the screen with the MAC address of network adapters 1-3 recorded earlier. See *fig. 11-52* below for an example, based on the MAC addresses of my IPS virtual machine.



11-52: Ubuntu has assigned our three virtual adapters the interface names ens32, ens33, and ens34. Below each interface name is a MAC address. By cross referencing those MAC addresses to the MAC addresses and the virtual switches attached to the IPS VM, we can determine which interface name maps to which network adapter, and confirm which virtual switch the interfaces are attached to. For instance, ens32, is the network adapter attached to

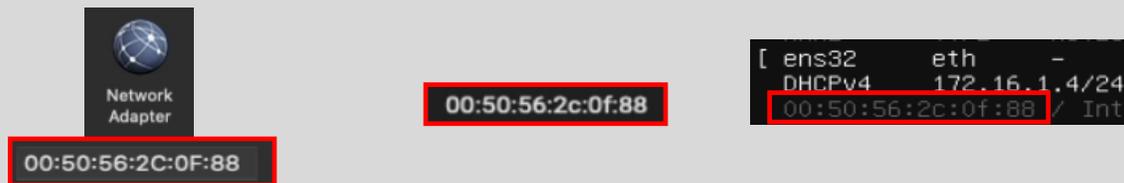
the *Host-only* network. ens33 is attached to the *VMnet2 (IPS1) network*, while ens34 is attached to the *VMnet3 (IPS2) network*.

Now that students are aware of which interface corresponds to which network segment, the next step is ensuring that the interface connected to the *Host-only* (e.g., the LAN network in pfSense) is the only interface that has an IP address assigned. In section 11.5.2, *fig. 11-40*, students created a static DHCP reservation for the IPS VM using the MAC address of the network adapter attached to the *Host-only* network, and assigned it the IP address 172.16.1.4. In *fig. 11-52* above, the interface ens32 has the IP address 172.16.1.4. This confirms students created the static DHCP allocation correctly on the pfSense WebConfigurator, and that ens32 is the interface connected to the *Host-only* network. If the network adapter attached to the LAN network does not have the correct IP address, there is a good chance that the static DHCP mapping for the IPS virtual machine is incorrect. Take a look at the side bar discussion, *Reservation for One*, for some troubleshooting recommendations.

Reservation for One

If you're here, that means that the network interface attached to the LAN/Management network didn't get the IP address 172.16.1.4. Similar to the *What Reservation?* Sidebar discussion for the SIEM VM, you'll want to check a few things:

- Check the MAC address of the network adapter attached to vmnet2
- Visit *Services > DHCP Server* and Check the Static DHCP Mappings of the *LAN* interface, particularly, the entry for the IPS VM
- Compare the MAC address of the previous two locations with the MAC addresses presented on the *Network Connections* screen. You should already know which interface name maps to which MAC address and network segment. In my case this was the interface ens32
- Make any corrections to the static DHCP allocation, then restart the IPS VM. Make your way back to the *Network connections* screen, and confirm that the correct interface was assigned the correct IP address.



11-53: Just like with the SIEM VM, compare the MAC address of the network adapter attached to vmnet2. Compare that to the MAC address used to create a static DHCP mapping for the IPS VM on the LAN interface of the pfSense webConfigurator. If they don't match, correct the static DHCP mapping entry, then restart the IPS VM. Determine which interface was assigned the IP address 172.16.1.4.

The final step on the *Network connections* screen is to disable the remaining network interfaces. The interfaces connected to the IPS1 and IPS2 port groups (In the illustrations provided, these are the interfaces *ens33* and *ens34*) should never receive an IP address. The lab environment, and IPS software we'll be using does not require these interfaces to have IP addresses, so we want to take advantage of that. Students may have notice that the interface connected to the VMnet3 (*IPS2*) network (*ens34*) doesn't have an IP address assigned, instead displaying the status: *disabled autoconfiguration failed*. Disregard this error message and follow the instructions below. Substitute the interface names *ens33* and *ens34* as necessary:

- Using the arrow keys, Highlight one of the other remaining interfaces. In my case, I chose to highlight *ens33*. Hit enter, and a dialogue box pops up.
 - Highlight the option *Edit IPv4*, and hit enter.
 - A new dialogue box appears titled *Edit ens33 IPv4 configuration*, with a single drop-down option highlighted, titled *IPv4 Method*. Hit enter again, and a list of choices appear. Use the arrow keys to select the option *Disabled*, and hit enter.
 - Use the arrow keys to highlight the option *Save*, and hit enter.
- Optional: Repeat the process again, only this time, Select *Edit IPv6*. By default, IPv6 should already be set to disabled, so this should not be necessary, but it is important to ensure these interfaces never receive an IPv4 or IPv6 address.
 - When finished, exit the *Edit ens33 IPv6 configuration* dialogue box.
- Repeat this process for the final interface. In my case, *ens34*. Disable the IPv4 Configuration (in my case, it was already set to *Disabled*) and confirm that the IPv6 configuration is already *Disabled*.

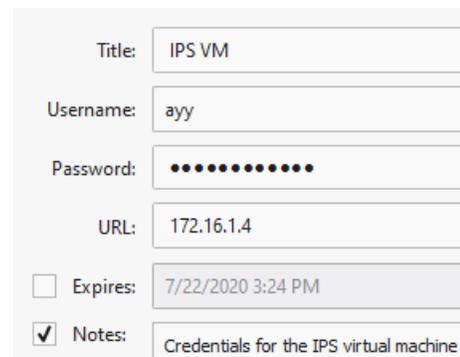
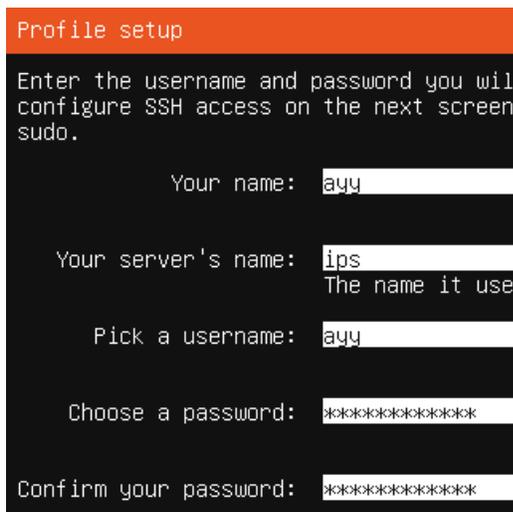
The end result should be one interface with the IP address 172.16.1.4, and two disabled network interfaces. Students can refer to *fig. 11-54* below for assistance. When finished, use the arrow keys to highlight *Done*, and hit enter to continue.



11-54: ens32 is the interface attached to the LAN network (*Host-only* network), and should be the only interface with an IP address. ***Disable the other interfaces. They should never be assigned an IP address.***

The rest of the installation process for the IPS VM should be identical to the SIEM VM:

- On the *Configure proxy* screen, set the *Proxy address* to `http://172.16.1.1:3128`
- Accept the default archive mirror (or an alternative, if required) on the *Configure Ubuntu archive mirror* screen
- Accept the default settings on the *Guided storage configuration*, and *Storage configuration* screens. Select *Continue* on the *Confirm destructive action* dialogue pop-up
- Fill out the *Profile setup* screen, ensuring that the *Your server's name* input box is set to *ips*. Remember to document the username and password you create and store it in your preferred password manager

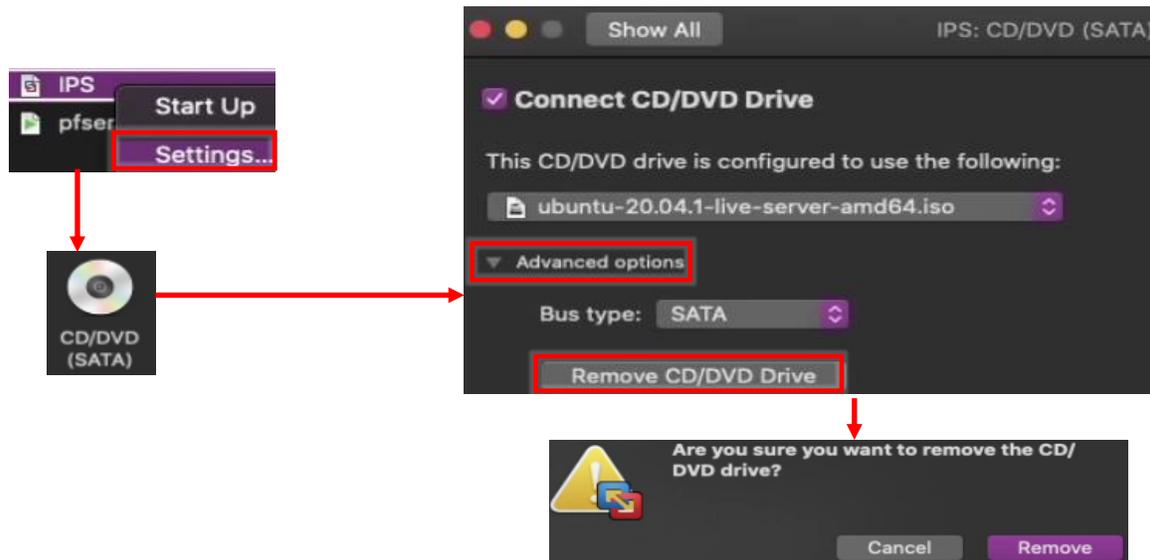


11-55: The *Profile setup* screen for the IPS virtual machine is, quite literally, the only other screen aside from the *Network connections* screen that differs from the installation process used on the SIEM VM. As always, save your username and password to a password manager!

- On the *SSH Setup* screen, be sure to select *Install OpenSSH server*
- On the *Featured Server Snaps* screen, select *Done* and hit enter to move on to the *Installation complete* phase
- Once the installation has finished, use the *Power off* option from the virtual console's *Actions* menu to shut down the virtual machine, just like the SIEM VM.

11.5.3.5 Additional Virtual Machine Settings – IPS VM

Now that Ubuntu Server is installed on the IPS VM, all that is left is to remove the *CD/DVD Drive (SATA)* virtual hardware. The process is identical to the one used on the SIEM virtual machine – Open the IPS VM's *Settings* menu, click on *CD/DVD Drive (SATA)*, *Advanced options*, then click the *Remove CD/DVD Drive* button. Close the menu window when this task is complete.



11-56: Access the IPS VM's *Settings* menu and *Remove the CD/DVD (SATA) Device*, then close the menu.

11.5.3.6 Booting the IPS VM for the first time

Start up on the IPS VM, and once Ubuntu has finished starting up and performing its first-time boot routines, log in with the username and password assigned on the *Profile setup* screen during the install. Just like with the SIEM VM, students will run the following three commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output from the `ip -br a` command will differ slightly, because the IPS VM has more network interfaces than the SIEM VM, but aside from that, the output from `nslookup` and `curl` should be more or less identical to the output of these commands from the SIEM VM. See fig. 11-57 below for an example on what the output of these commands should look like.

```

Ubuntu 20.04 LTS ips tty1

ips login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@ips:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
ens32             UP              172.16.1.4/24 fe80::250:56ff:fe34:b764/64
ens33             DOWN
ens34             DOWN
ayy@ips:~$ nslookup www.google.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.8.196
Name:   www.google.com
Address: 2607:f8b0:4009:815::2004
ayy@ips:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Thu, 23 Jul 2020 17:21:51 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Thu, 23 Jul 2020 17:21:51 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-23-17; expires=Sat, 22-Aug-2020 17:21:51 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=No2uEqnF70q9zD3pzs0rY1b9m4o1HDDzP4BzZ1ULDM2ia7uXqWv97cWdZN0fc2JxijI_BXyxhRfuF2EEvFV50ssKkaJRIZPxm4TbIdfzAihP6aW6FsTqHu6Kif6j75q06iuFFU-UP0oA73r0ytPyD314nvxBKnu1_rqEmla0Sic; expires=Fri, 22-Jan-2021 17:21:51 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"

```

11-57: Just like with the SIEM VM, students will log in to the IPS virtual machine and run a couple of network diagnostic commands. The output from `curl` and `nslookup` commands should be more or less identical to the output on the SIEM VM, but the `ip -br a` command will produce a few more lines of content. Ignoring the `lo` (loopback) interface, there should be three interfaces. Only one of them should have the status of `UP`. That interface should be the interface assigned to the LAN/Host-only network, with the IP address `172.16.1.4`.

After running these commands to confirm the IPS VM has been assigned the proper IP address, can resolve hostnames, and has HTTPS connectivity, run the commands:

```

sudo su -
apt-get update
apt-get -y dist-upgrade
init 6

```

In order to become the root user, install updates on the IPS VM (and confirm the Squid proxy server is proxying the IPS VM's HTTP requests), then reboot after the system is done installing those updates.

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [306 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [114 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [7612 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [136 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [4792 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Fetched 889 kB in 1s (1269 kB/s)
Reading package lists... Done
root@ips:~# apt-get -y dist-upgrade_
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@ips:~# init 6

```

11-58: These commands are identical to the ones students ran on the SIEM virtual machine, and the serve the same purpose for the IPS VM: become the root user, check for updated packages, install those updates, then reboot the system.

Note: If you're having problems with your apt-get commands failing, refer back to the sidebar conversation on pp. 369-370, [*Help! My apt-get commands are failing!*](#) For further guidance. Students can follow the exact same steps laid out for the SIEM VM to troubleshoot the problem.

11.5.3.7 Installing Kali Linux on the kali VM

Now that the SIEM and IPS virtual machines are out of the way, next up is the kali VM. Start up the VM to begin the boot process and connect to the virtual console. A boot menu appears with a number of options. Using the arrow keys, highlight *Install* and hit enter.

Similar to the Ubuntu installer, the first screen, titled *Select a language*, asks users to choose the language they want to use for their installation. The default setting is *English*, use the arrow keys to highlight another language as necessary, then hit enter. The next screen, *Select your location*, asks users to choose what country, territory or area in which they are located. This screen defaults to *United States*. Use the arrow keys to change this value as necessary, and hit enter to continue. Next up is the *Configure the keyboard* screen, that asks the user what keymap to use for their installation. The default setting is *American English* and can be changed with the arrow keys. After highlighting a keymap, hit enter to continue.

The installer begins loading other phases and components it will need later. Afterwards, it will attempt to get an IP address. The pfSense DHCP server should give it an IP address through the OPT1 DHCP server, but students will not be able to confirm if the IP address 172.16.2.2 was

correctly assigned until after the operating system is installed. The next screen, titled *Configure the network*, prompts users to enter a hostname for the system. Students should use the default hostname *kali*. Hit the enter key to continue to the next screen that prompts for a domain name. Again, students may hit enter and accept the default, *localdomain*.

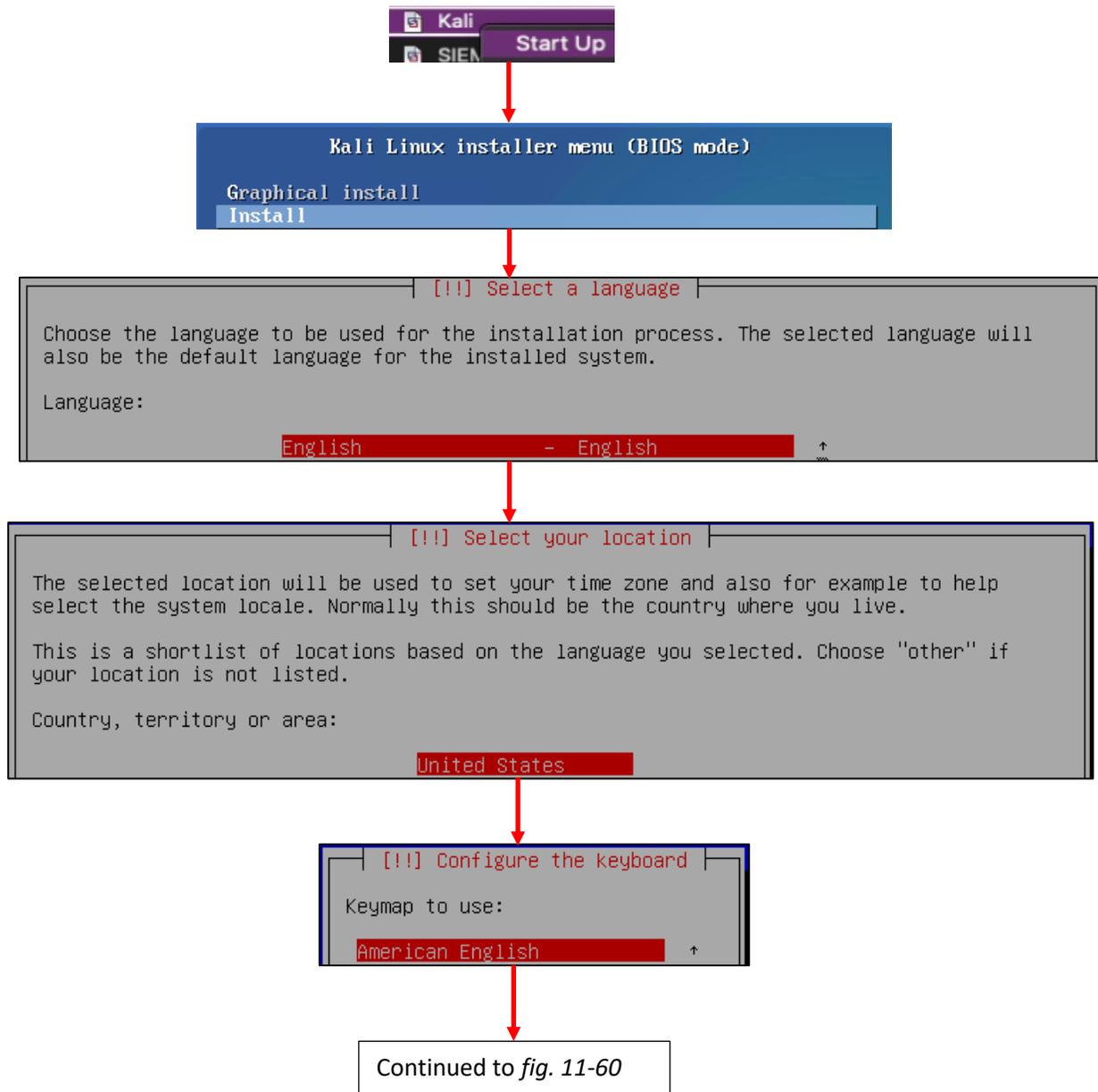
The *Set up users and passwords* screen appears. The first window asks for the full name of the user to be created. Type in the full name of the user account, and hit enter to continue to the next screen, that prompts for a username students will use to log in to the system. After typing in a username, hit enter to be prompted to create a password for this account. After hitting enter again, you'll be prompted to enter the same password again to confirm your choice. Just like with the SIEM and IPS virtual machines, be sure to save the username and password to your preferred password manager.

Next up is the *Configure the clock* dialogue. The installer will reach out to its preferred NTP servers to get the current time, then ask the user to select the time zone in which they are located. Use the arrow keys to choose your time zone, and hit enter to continue.

The *Partition disks* screen appears and asks users to select a partitioning method. Highlight the selection *Guided – use entire disk*, and hit enter. Users are then prompted to select the disk to partition. Since there is only a single virtual disk for the kali VM, hit enter to proceed. The next screen prompts students to select the partitioning scheme. Highlight the option *All files in one partition (recommended for new users)* and hit enter. Users are asked to confirm their choices on the next screen. Highlight the option *Finish partitioning and write changes to disk*, and hit enter. One final pop-up appears to annoy you, asking if students are sure they want to proceed, highlight *<Yes>* to confirm your choices, and press enter to continue.

The installer proceeds and begins installing the base operating systems components to the newly partitioned disk. After a moment or two, a window labeled *Software selection* appears. As the name implies, this screen allows users to pick additional software packages to install. Accept the default selections by pressing the tab key to highlight *<Continue>*, and hitting enter the next portion of the installer Retrieves and installs the requested packages. This portion of the installation may take some time, depending on internet speed and virtual machine performance.

After some time has passed a new prompt appears, labeled *Install the GRUB boot loader on a hard disk*, asking if users want to install the GRUB boot loader. This is a necessary component in order to boot the virtual machine, so highlight *<Yes>*, and press the enter key to continue. The next screen asks what partition to install the boot loader to. Seeing as how there is only one partition available, highlight it, and hit enter to proceed. After a moment or two passes, students are prompted to remove the installation media, and reboot the virtual machine to complete the installation. Just like with the SIEM and IPS virtual machines, *Turn Off* the virtual machine, then close the virtual console.



11-59: The first screens have users select their preferred language, location, and keyboard keymap.

Continued from *fig. 11-59*

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

kali

<Go Back> <Continue>

Domain name:

localdomain

<Go Back> <Continue>

Continued to *fig. 11-61*

11-60: The next few screens configure some of the network settings. Students are prompted to enter a hostname and domain name. Students should use the default hostname of *kali*, and default domain name of *localdomain*.

Continued from *fig. 11-60*

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

ayy lmao

<Go Back> <Continue>

Username for your account:

ayy

<Go Back> <Continue>

Choose a password for the new user:

[] Show Password in Clear

<Go Back> <Continue>

Re-enter password to verify:

[] Show Password in Clear

<Go Back> <Continue>

Continued to *fig. 11-62*

Title:	kali VM
Username:	ayy
Password:	*****
URL:	172.16.2.2
<input type="checkbox"/> Expires:	7/24/2020 11:35 AM
<input checked="" type="checkbox"/> Notes:	credentials for the kali VM

11-61: Similar to the *Profile Setup* screen in the Ubuntu installer, the Kali Linux installer features a series of prompts to create a user account for the system. Be sure to save the credentials to your preferred password manager when finished.

Continued from *fig. 11-61*

[!] **Configure the clock**

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

Eastern

[!!] **Partition disks**

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk

Select disk to partition:

SCSI3 (0,0,0) (sda) - 85.9 GB VMware, VMware Virtual S

Partitioning scheme:

All files in one partition (recommended for new users)

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI3 (0,0,0) (sda) - 85.9 GB VMware, VMware Virtual S

#1	primary	84.9 GB	f	ext4	/
#5	logical	1.0 GB	f	swap	swap

Undo changes to partitions
Finish partitioning and write changes to disk

Write the changes to disks?

<Yes> <No>

Continued to *fig. 11-63*

11-62: After setting the time zone, students will have to configure the partitioning scheme for the install. The highlighted options above should be selected by default. If not, use the arrows to select them, and press enter to continue.

Continued from *fig. 11-62*

```
[!] Software selection

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

[*] Desktop environment [selecting this item has no effect]
[*] ... Xfce (Kali's default desktop environment)
[ ] ... GNOME
[ ] ... KDE Plasma
[*] Collection of tools [selecting this item has no effect]
[*] ... top10 -- the 10 most popular tools
[*] ... default -- recommended tools (available in the live system)
[ ] ... large -- default selection plus additional tools

<Continue>
```

Continued to *fig. 11-64*

11-63: On the *Software selection* screen, press the tab key to highlight *<Continue>*, and accept the default packages.

Continued from *fig. 11-63*

```
[!] Install the GRUB boot loader on a hard disk

It seems that this new installation is the only operating system on this computer. If so,
it should be safe to install the GRUB boot loader to the master boot record of your first
hard drive.

Warning: If the installer failed to detect another operating system that is present on
your computer, modifying the master boot record will make that operating system
temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

<Go Back> <Yes> <No>
```

```
[!] Install the GRUB boot loader on a hard disk

You need to make the newly installed system bootable, by installing the GRUB boot loader
on a bootable device. The usual way to do this is to install GRUB on the master boot
record of your first hard drive. If you prefer, you can install GRUB elsewhere on the
drive, or to another drive, or even to a floppy.

Device for boot loader installation:

Enter device manually
/dev/sda

<Go Back>
```

```
[!] Configuring common

Installation complete

Installation is complete, so it is time to boot into your new system. Make sure to remove
the installation media, so that you boot into the new system rather than restarting the
installation.

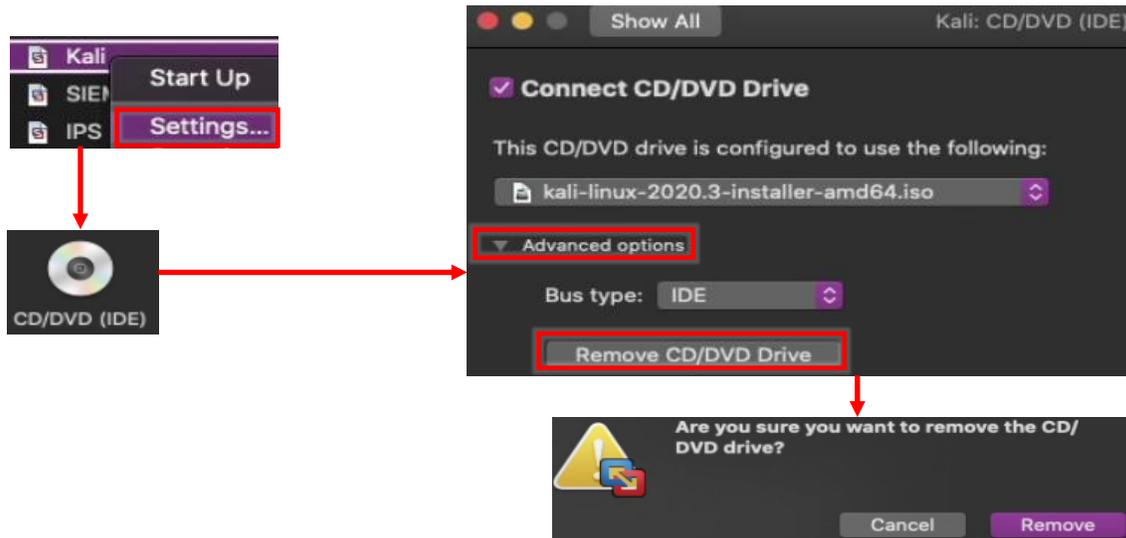
<Go Back> <Continue>
```

```
Virtual Machine
Linux is running
Suspend
Restart
Shut Down
```

11-64: The final steps of the installation process. Install the GRUB boot loader to the only available disk on the system (this should be `/dev/sda`), wait for the screen labeled *Configuring common* to appear, then *Shut Down* the virtual machine.

11.5.3.8 Additional Virtual Machine Settings – kali VM

By this point, it should already be established routine that once students are finished installing the operating system on their virtual machine, the next step is to remove the CD/DVD Drive, and Kali is no exception. Open up the Kali VM's *Settings* menu, click on *CD/DVD Drive (IDE)*, *Advanced options*, then click the *Remove CD/DVD Drive* button. Close the menu window when this task is complete.



11-65: Access the Kali VM's *Settings* menu, *Remove the CD/DVD (IDE) Device*, then click *OK* to exit.

11.5.3.9 Booting the kali VM for the first time

With those last-minute virtual machine settings applied, *Start Up* the Kali virtual machine. After a moment or two passes, students will be greeted with a graphical interface, asking for a username and password to log in. Enter the username and password supplied during the operating system install, and click *Log In* to continue.

On the top of the graphical user interface, there should be a menu bar with a few icons displayed. One of those icons is a small black window. Click on that icon to open a terminal session on the kali VM. With the terminal window open, run the same three commands that we ran on the SIEM and IPS virtual machines in order to confirm network connectivity is working as intended:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output of `ip -br a` should confirm that only a single interface (again, ignoring the `lo` interface) is installed on the system. That interface should have the IP address `172.16.2.2`. As with the SIEM and IPS virtual machines, if this is not the case, students should compare the MAC address of network adapter on the kali VM to the MAC address of the static DHCP mapping made on pfSense. **Make sure the mac addresses match, and that the mapping was created on the OPT1 interface.**

As with the SIEM and IPS VMs, nslookup confirms the ability of the kali VM to resolve hostnames through DNS, and the curl command verifies that the VM can make outbound internet connections over HTTPS. The output of these commands should be similar to the output displayed in *fig. 11-66* below.

While Kali Linux is slightly different from Ubuntu, we can still use *most* of the same commands utilized on the SIEM and IPS virtual machines to become root, check for updates, then reboot the system. **Run these commands in this exact order:**

```
sudo su -
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
apt-get update
apt-get -y dist-upgrade
init 6
```

Students may have noticed two new commands have been added here:

```
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
```

These commands are responsible configuring the apt package manager to use our HTTP proxy at 172.16.2.1:3128 on the *OPT1* interface of the pfSense VM. This is done by running the echo command, and redirecting its output (the > symbol) to the file /etc/apt/apt.conf.d/99local (a configuration file that the package manager will read when we run apt-get later). The second command, cat /etc/apt/apt.conf.d/99local, reads the contents of the file. If the output from the cat command reads:

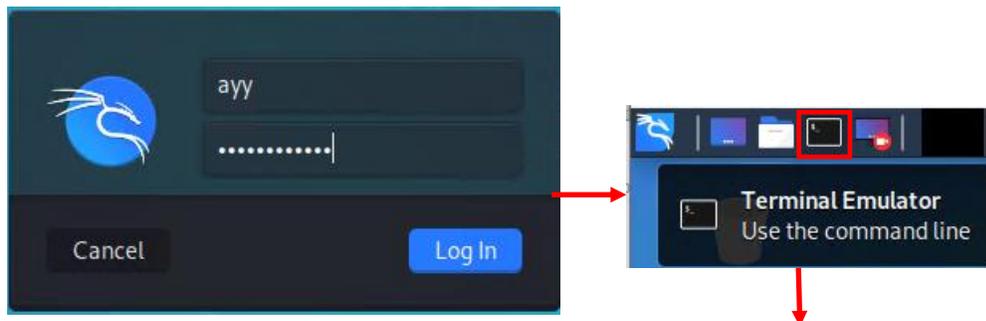
```
Acquire::http::Proxy "http://172.16.2.1:3128";
```

Then that means apt was successfully configured to use the HTTP proxy. If the output from the cat command displays anything else, then students should re-enter the echo command.

Note: If most of these commands look familiar, it's because they're very similar to the troubleshooting commands I recommended in the sidebar discussion [*Help! My apt-get commands are failing!*](#) (pp. 369-370) for the SIEM and IPS virtual machines. There are a few key differences with the kali VM to be aware of, but for the most part, the troubleshooting steps laid out are the same as the steps I laid out in this section. Here are the key differences to be aware of:

- Make absolutely sure you are redirecting the output of the echo command to the file /etc/apt/apt.conf.d/99local. **It must be that exact file, in that exact location.**
- The kali VM doesn't need the second line, Acquire::https::Proxy "http://172.16.2.1:3128";
- Make absolutely sure to specify http://172.16.2.1:3128 as the proxy address for the kali VM.

After running these commands to configure the package manager, students should be able to run the remaining commands just like on the SIEM and IPS virtual machines. Bear in mind that Kali Linux is subject to frequent updates, and that some of those updates can be quite large. This means that depending on the performance of the Kali VM, and internet connection speeds, downloading and installing updates may take some time to complete.



```

ayy@kali:~$ ip -br a
lo          UNKNOWN          127.0.0.1/8  ::1/128
eth0       UP                    172.16.2.2/24 fe80::581c:6ed2:259e:5cb/64
ayy@kali:~$ nslookup www.google.com
Server:      172.16.2.1
Address:     172.16.2.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:804::2004
ayy@kali:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 24 Jul 2020 19:55:47 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 24 Jul 2020 19:55:47 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-24-19; expires=Sun, 23-Aug-2020 19:55:47 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=rC1q-094PKdmAIZC2ajgCkpdGrGdulzdaxnJR2Cui-HYKBgbjzh_qCz6G5tJYoetE_Uc7rscR53x4Gri7HE3k_gu9h2BKh6etyF0hGD0iat3FF22oe-4VngjLFAdGEY3XTecVHB8iJ5Qw2qUVjmmN7oZGeRUSj1mXJ8ulaLkRY; expires=Sat, 23-Jan-2021 19:55:47 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@kali:~$ sudo su -

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for avv:
root@kali:~# echo 'Acquire::http::proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
root@kali:~# cat /etc/apt/apt.conf.d/99local
Acquire::http::proxy "http://172.16.2.1:3128";
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@kali:~# init 6

```

11-66: Login to the kali VM, configure the apt package manager to use the SQUID HTTP proxy on *OPT1* of the pfSense VM. Afterwards, install the latest operating system updates, then reboot the virtual machine.

11.5.4 Metasploitable 2

The Metasploitable 2 VM is covered separately from the other VMs students have created because it's a special case. Technically the VM is already created, and all that needs to be done is to "register" it with VMware Fusion. However, there are a number of small configuration tasks that need to be covered both before and after registering the virtual machine.

11.5.4.1 Registering the Metasploitable 2 VM

Note: Before we begin, please make sure that you have downloaded the Metasploitable 2 VM from Sourceforge, and that you've decompressed the `metasploitable-linux-2.0.0.zip` file. For guidance, check out Chapter 1, [section 1.8 \(Using Compression Tools](#), pp. 33-35). you'll need access to the entire `Metasploitable2-Linux` directory to perform the tasks ahead.

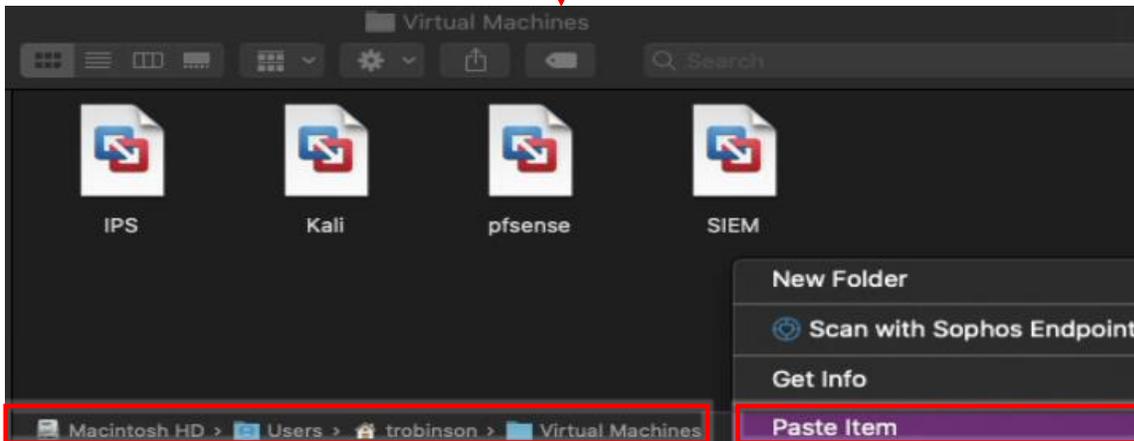
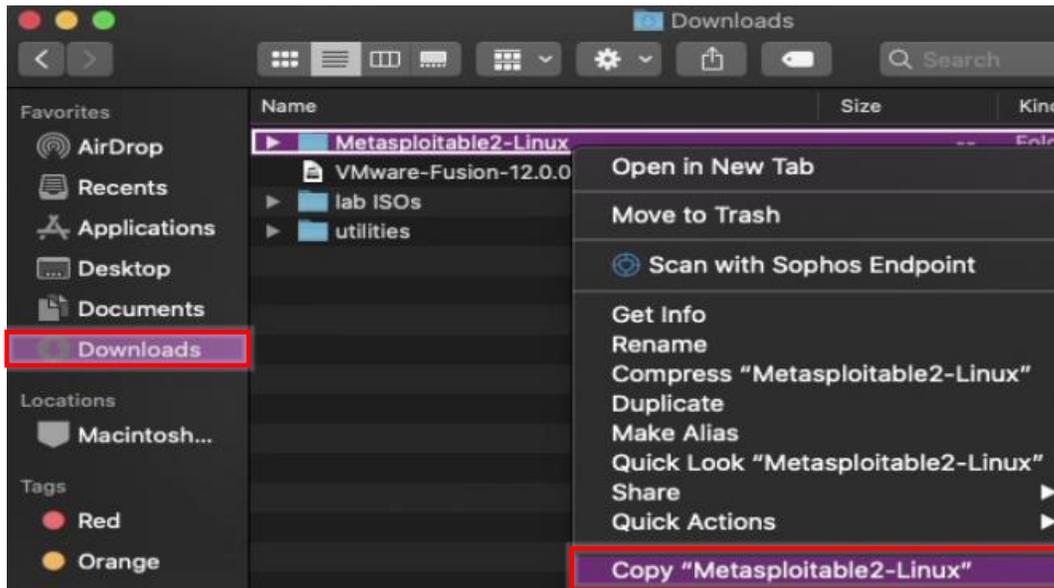
Back in [section 11.4.1](#) (pp. 318-321), we discussed that VMware Fusion stores virtual machines in:

```
/Users/[username]/Virtual Machines.localized.
```

As always, students will need to replace the `[username]` placeholder in the path above with their actual username they log in with.

It was also mentioned that in the *Finder* file browser, this folder appears as "Virtual Machines".

The first task students will need to perform is copying the entire *Metasploitable2-Linux* directory to that directory. To perform this task, MacOS provides the *Finder* file browser for navigating the file system and moving files from one location to another. *Copy* the `Metasploitable2-Linux` directory from one folder, and *Paste* it into the `Virtual Machines` folder.



11-67: Using *Finder* to *Copy* and *Paste* the *Metasploitable2-Linux* directory to the *Virtual Machines* directory is probably the fastest and easiest way to accomplish this task.

Alternatively, students can use the terminal (e.g., *iTerm/iTerm2*) and the `cp -r` command instead.

The `cp`, or "copy" command is used to copy files from one location to another. The recursive `-r` option is required for copying folders and all their contents – it says "copy this entire folder and all of its content to the location specified." Assuming the *Metasploitable2-Linux* directory is located in the current user's *Downloads* directory, use the following command to copy it to the *Virtual Machines.localized* directory:

```
cp -r /Users/[username]/Downloads/Metasploitable2-Linux /Users/[username]/Virtual\
Machines.localized/
```

Replace the `[username]` portion of the command above with your username. For example, the username on my system is "trobinson":

```
cp -r /Users/trobinson/Downloads/Metasploitable2-Linux /Users/trobinson/Virtual\
Machines.localized/
```

Students may need to adjust the command according to where they stored the Metasploitable2-Linux folder. **Additionally, please be aware that the backslash (\) in Virtual\ Machines.Localized in the cp command is required.** Students can confirm the successful transfer of the Metasploitable2-Linux directory by running the command:

```
ls -al /Users/[username]/Virtual\ Machines.localized
```

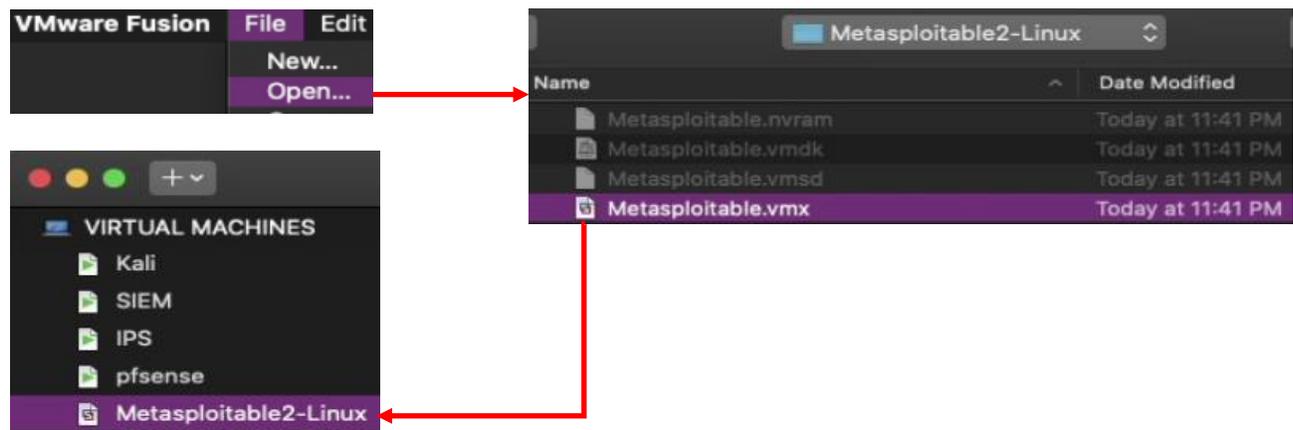
Once again, replace [username] with the username students use to log in to the host (e.g., "trobinson"):

```
ls -al /Users/trobinson/Virtual\ Machines.localized
```

```
trobinson@trobinsons-MacBook-Pro Downloads % pwd
/Users/trobinson/Downloads
trobinson@trobinsons-MacBook-Pro Downloads % cp -r Metasploitable2-Linux /Users/trobinson/Virtual\ Machines.localized
trobinson@trobinsons-MacBook-Pro Downloads % ls -al /Users/trobinson/Virtual\ Machines.localized
total 0
drwxr-xr-x  8 trobinson  staff   256 Nov 12 14:52 .
drwxr-xr-x+ 25 trobinson  staff   800 Nov 12 10:50 ..
drwxr-xr-x 10 trobinson  staff   320 Sep 11 02:48 .localized
drwxr-xr-x@ 37 trobinson  staff  1184 Nov 12 10:51 IPS.vmwarevm
drwxr-xr-x@ 13 trobinson  staff   416 Nov 12 13:50 Kali.vmwarevm
drwx-----@ 7 trobinson  staff   224 Nov 12 14:52 Metasploitable2-Linux
drwxr-xr-x@ 19 trobinson  staff   608 Nov  9 20:36 SIEM.vmwarevm
drwxr-xr-x@ 15 trobinson  staff   480 Nov 12 10:51 pfsense.vmwarevm
```

11-68: Alternatively, the cp command can be used for copying files and folders, instead. Pay attention to the backslash in Virtual\ Machines in the cp -r command. The backslash is used to escape characters that have special meaning on the command line. In this case, it is necessary to escape the space character between Virtual and Machines in the directory path.

With the Metasploitable2-Linux directory moved, open VMware Fusion, and select *File > Open* from the navigation menu. This opens a *Finder* file browser. Navigate to the newly relocated Metasploitable2-Linux directory, and select the Metasploitable.vmx file. Upon selecting it, a new entry titled Metasploitable2-Linux should appear in the *Virtual Machine Library* window.

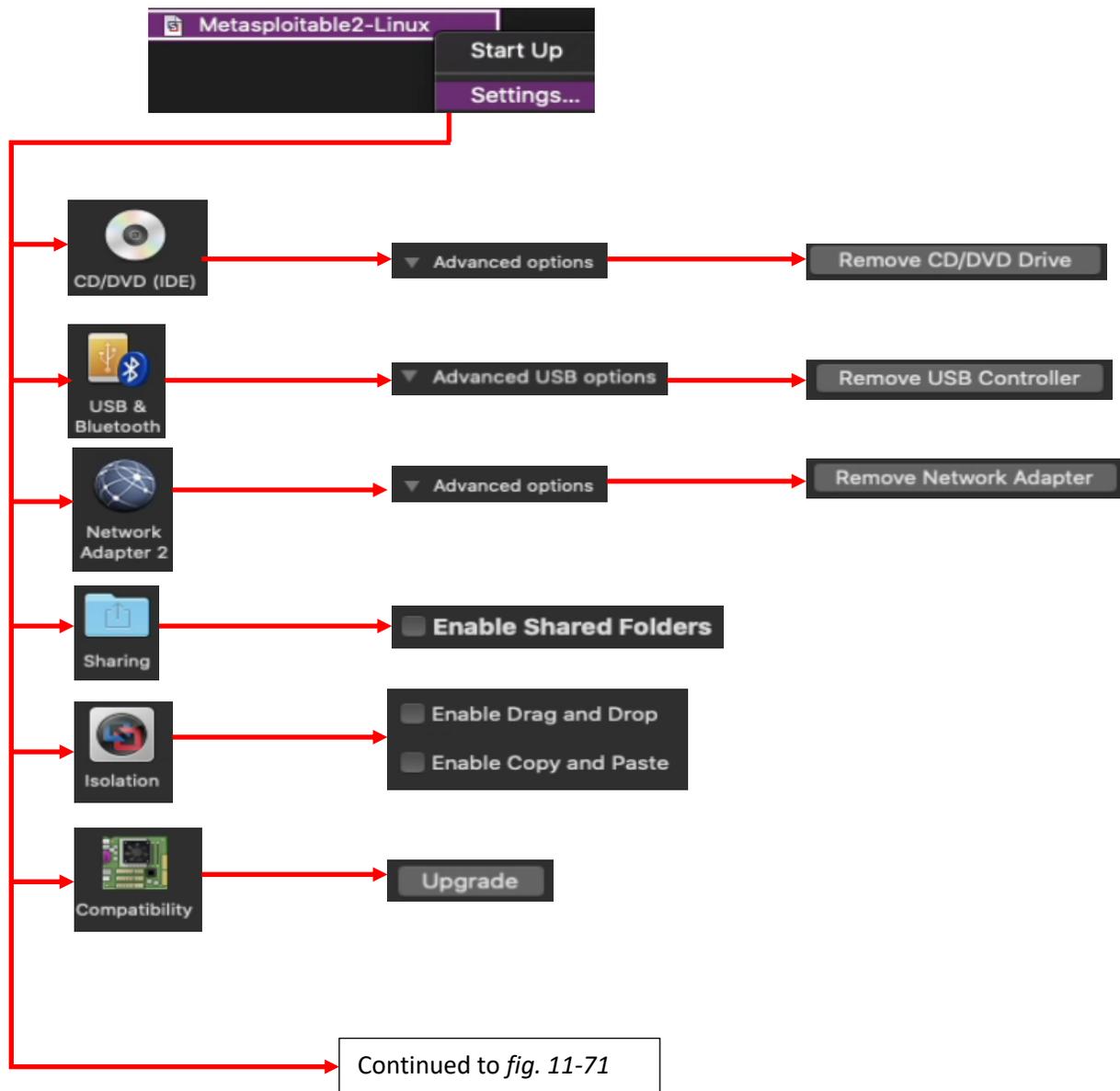


11-69: Open VMware Fusion, and from the navigation menu, select *File > Open*. Use the file browser to navigate to /Users/[username]/Virtual Machines/Metasploitable2-Linux and select the Metasploitable.vmx file. This will add an entry titled Metasploitable2-Linux to the *Virtual Machine Library* window.

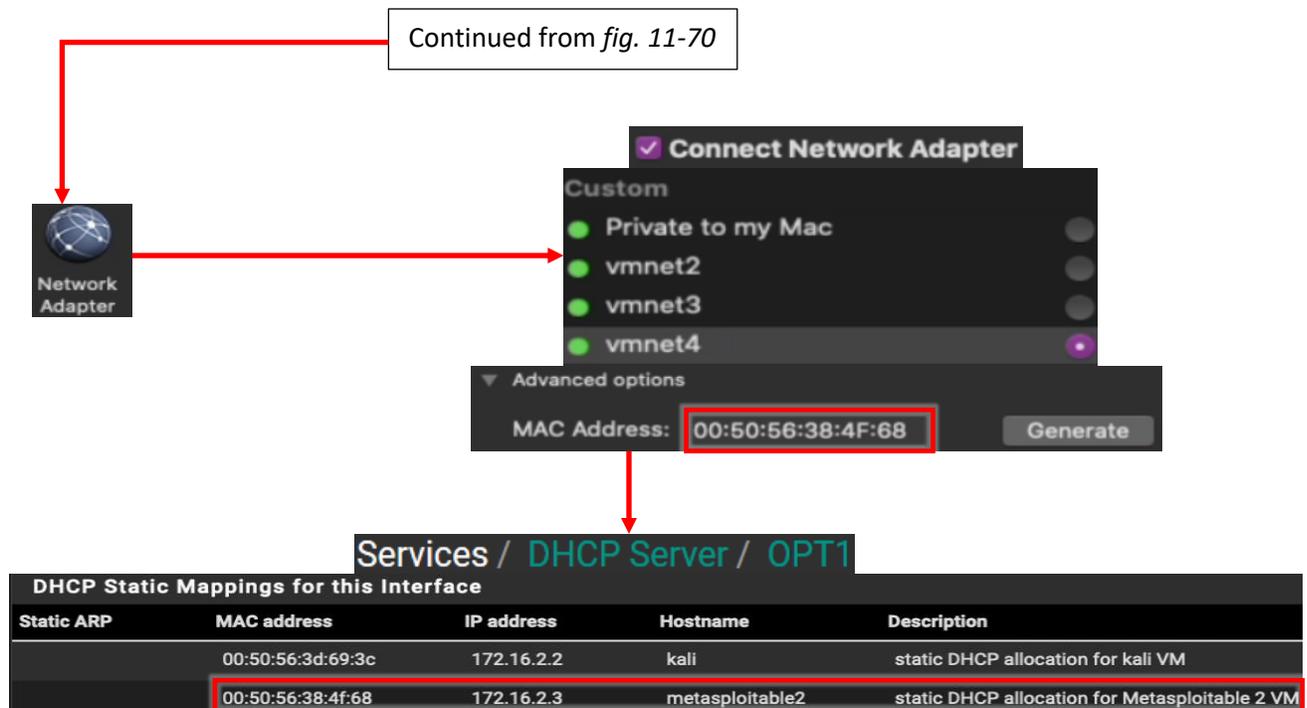
11.5.4.2 Edit Metasploitable 2 Virtual Machine Settings

Right-click on the Metasploitable2-Linux entry in the *Virtual Machine Library Window*, and select *Settings*. There are a lot of options that need to be modified here, but the nice thing about Metasploitable 2 is that the operating system is pre-installed, so students will be able to perform all of the necessary edits in one go. Perform the following configuration changes:

- Remove the *CD/DVD (IDE)*, *USB Controller*, and *Network Adapter 2* virtual hardware devices. As a general reminder, you'll need to click *Advanced options* to expose the *Remove CD/DVD Drive*, *Remove Network Adapter*, and *Remove USB Controller* buttons.
- Navigate to *Sharing* and ensure the *Enable shared folders* checkbox is unchecked
- Navigate to *Isolation* and uncheck both the *Enable Drag and Drop* and *Enable Copy and Paste* checkboxes
- Navigate to *Compatibility*. Click the *Upgrade* button to upgrade the virtual machine's hardware to latest version (as of writing, the current hardware version is 18)
- Navigate to the *Network Adapter* menu. Ensure the *Connect Network Adapter* checkbox is checked. In the pane below, navigate to *Custom*, and select the *VMnet4* radio button. Click *Advanced options*, and in spite of there already being MAC address in the input box, click the *Generate* button and document the newly generated MAC address that appears.
- Open a session to the pfSense WebConfigurator, navigate to *Services > DHCP Server > OPT1* and create a static DHCP mapping for the newly recorded MAC address of the metasploitable 2 VM, assigning it the IP address 172.16.2.3. **It is extremely important that the Metasploitable 2 VM always gets the IP address 172.16.2.3.** Apply your changes.



11-70: Open the Metasploitable 2 VM's *Settings* menu. While there are a lot of configuration options that need to be modified, most of these options should be routine by now. Begin by removing the *CD/DVD Drive*, *USB Controller*, and *Network Adapter 2* devices. Next, under *Sharing*, confirm that *Enable Shared Folders* is disabled. Then, navigate to *Isolation* and uncheck both the *Enable Drag and Drop*, and *Enable Copy and Paste* checkboxes. Afterwards, select *Compatibility*, and click the *Upgrade* button on the screen. This should automatically upgrade the VM's virtual hardware to the latest version available.



11-71: The last thing to do in the *Settings* menu involves configuring the remaining *Network Adapter*. Ensure that the *Connect Network Adapter* checkbox is checked, then in the pane below, select the *vmnet4* radio button under *Custom*. Finally, open the *Advanced options*, then click *Generate* to assign a new MAC address. Record this MAC address, then log into the pfSense WebConfigurator. With the MAC address students just generated and recorded, create a new static DHCP mapping for the Metasploitable 2 VM on the *OPT1* interface. **Assign the Metasploitable 2 VM the IP address 172.16.2.3. This is extremely important.**

11.5.4.3 Metasploitable 2 Test Run

Start Up on the Metasploitable 2 VM, and connect to its virtual console. A whole bunch of text will scroll by as the VM goes through the boot process. After some time has passed, students should be greeted with a login prompt. The default credentials for metasploitable 2 are the username and password combination of *msfadmin/msfadmin*. Upon logging in, run the command `ifconfig -a`, and confirm that the interface `eth0` appears. **Record the contents of the field labeled `Hwaddr`, the MAC address of `eth0`.** Confirm it matches the MAC address of the static DHCP mapping students just created for the Metasploitable 2 VM. When finished, type `exit` to log out of the virtual machine.

```
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Sun May 20 15:50:42 EDT 2012 from 172.16.123.1 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:50:56:38:4f:68
          inet6 addr: fe80::250:56ff:fe38:4f68/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2178 (2.1 KB)
          Interrupt:17 Base address:0x2000

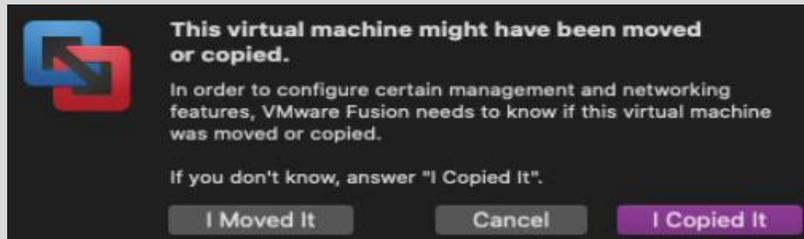
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27753 (27.1 KB)  TX bytes:27753 (27.1 KB)
```

Services / DHCP Server / OPT1
00:50:56:38:4f:68 172.16.23 metasploitable2

11-72: Power on the Metasploitable VM, and connect to its virtual console. Login with the username/password of msfadmin/msfadmin. Run the command `ifconfig -a`, record the HWaddr field for the eth0 interface. Confirm it matches the MAC address of the static DHCP mapping students just created for the Metasploitable 2 VM then type `exit` to log out.

Who Touched my VM?

When you first power on the Metasploitable 2 VM, you'll be greeted by a pop-up that demands to know whether you moved or copied the VM before it will start up. Click the button labeled *I Copied It* to proceed. This, may lead to another pop-up that demands you to install VMware tools. Click *OK*, ignore this pop-up and continue asserting your dominance over the machines.



11-73: Who touched Sasha? WHO TOUCHED MY VM?



11-74: This message pops up immediately after, insisting users install VMware tools on this VM. Show the machine legion how much you care by clicking *OK* (and optionally, the *Never show this dialog again* checkbox) and ignoring this message.

Why aren't we doing connectivity checks?

Some of you may be wondering why we aren't doing connection checks or any of the stuff we did we for the SIEM, IPS, or Kali VMs, like checking that the static DHCP allocation is working, or attempting to connect outbound. Well, that's because right now, the metasploitable 2 VM doesn't have an IP address at all. Don't worry, its intentional, and you'll be fixing this later. The reason metasploitable 2 doesn't have an IP address is that it's connected to the *VMnet3* (IPS2) network. While technically the *VMnet3* (IPS2) shares the same network subnet as *VMnet2* (IPS1), and logically it's all a part of the *OPT1* network, IPS2 is its own physical network segment, and entirely separate from the IPS1 network. **Without something to bridge connect the IPS1 and IPS2 networks together, the IPS2 network is entirely isolated.**

Remember the network diagram back in [chapter 6](#) (p. 58)? The *IPS2* network relies on the IPS virtual machine being fully configured and running either Snort or Suricata in AFPACKET bridging mode. No network bridge, no network connectivity. That means no IP address from the DHCP server, either. You can see this for yourself from the output of `ifconfig -a`. You'll be fixing this later when you install either Snort or Suricata to the IPS virtual machine in [chapter 17](#).

```
eth0      Link encap:Ethernet  HWaddr 00:50:56:38:4f:68
          inet6 addr: fe80::250:56ff:fe38:4f68/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2178 (2.1 KB)
          Interrupt:17 Base address:0x2000
```

11-75: `eth0` never got assigned an IPv4 address from DHCP because there is no physical connectivity between the IPS1 and IPS2 networks (*VMnet2* and *VMnet3*, respectively). We'll be solving this problem later when students install Snort or Suricata to the IPS virtual machine.

11.6 Snapshots

The next (and final) task for students to perform will be creating baseline virtual machine snapshots for the entire lab environment. Snapshots (sometimes referred to as checkpoints by other hypervisors) instruct the virtual machine's hypervisor to gather information about the VM's current state, and save it. Later on, if there is a problem with the virtual machine such as a malware infection, or a configuration problem that cannot be diagnosed, users can choose to restore the virtual machine to its state in the past, when the snapshot was initially created.

Snapshots can be created with virtual machines powered off, or while they are running, making them extremely versatile. VMware Workstation virtual machines can also have more than one checkpoint, with the only limit being disk space required to hold them. It's extremely important to note that **virtual machine snapshots are not a substitute for backups**. If students plan on running virtual machines with important data that they cannot afford to lose, snapshots are not a substitute for backing up important files and data.

In this section, students will walk through the process of creating a virtual machine snapshot for the pfSense VM. Afterwards, it will be left as an exercise to the students to repeat the process for the SIEM, IPS, kali, and Metasploitable 2 virtual machines. Once finished, students will be ready to move on with the configuration of their lab environment.

11.6.1 How to Create a Snapshot

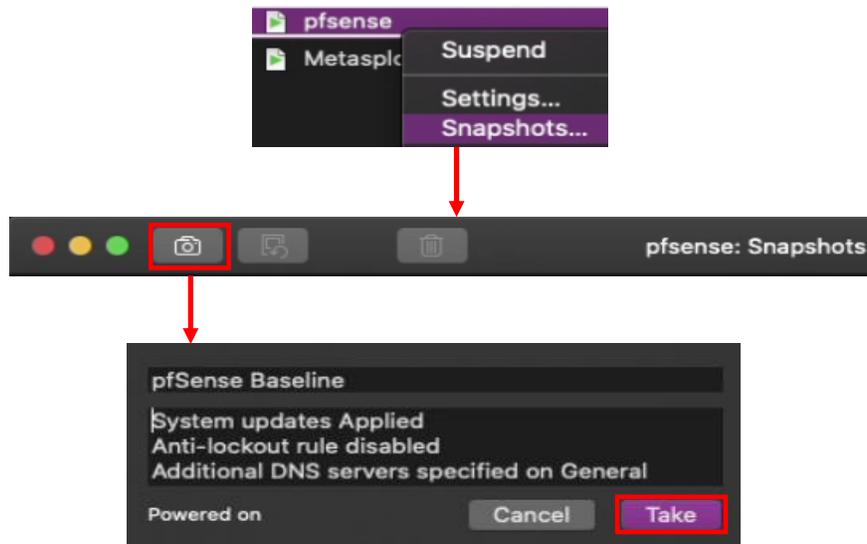
In the *Virtual Machine Library* window of the VMware Fusion interface, right-click on the pfSense listing. Select the option labeled *Snapshots*, and a window labeled *pfSense: Snapshots* will appear. Click the small icon in the upper left corner of the window that looks like a camera. A pop-up with two input boxes appears. In the first, smaller input box, enter a name for this snapshot that provides a brief description about the state of the virtual machine. The second input box can be used to provide more detailed information about the state of the VM. For example, I recommend entering the following *Name* and *Description* for the pfSense VM's first snapshot:

Name: pfSense Baseline

Description:

System updates Applied
Anti-lockout rule disabled
Additional DNS Servers specified on General Settings page.
DNS Server Override disabled
DHCP Server enabled, Static DHCP mappings applied
DNS Resolver service enabled for LAN and OPT1
Squid proxy service installed and enabled for LAN and OPT1
NTP enabled for LAN and OPT1
Firewall policy applied for WAN, LAN and OPT1

Once finished, click the *Take* button for Fusion to begin creating the snapshot.

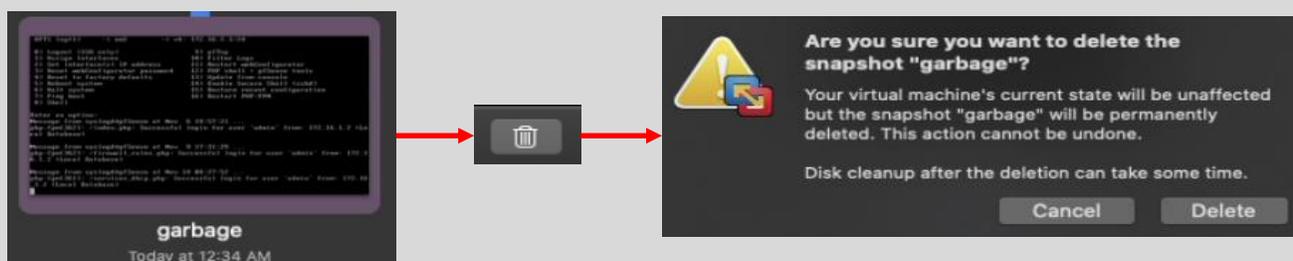


11-76: Right-click on the pfSense entry on the *Virtual Machine Library* window, and select *Snapshots*. In the new window that appears, click on the icon that looks like a camera. In the pop-up window that appears, enter a name and description for your snapshot, then click the *Take* button. Use descriptive names, and add detailed descriptions. Tip: students can add multiple lines to the bigger text box by using ctrl+enter to create new lines.

11.6.2 Restoring a Snapshot

In order to restore a virtual machine snapshot, right-click the target VM in from the list on the *Virtual Machine Library* window and select *Snapshots*. Once again, the window titled *[Virtual Machine Name]: Snapshots* will appear, complete with a gallery of all the snapshots created for that VM. Left click the desired snapshot to highlight, then click the icon in the upper left that feature a square with a circular arrow. A pop-up appears asking if the user wishes to save the virtual machine in its current running configuration before restore the snapshot. Select *Save* or *Don't Save* depending on whether or not there is a need to save the VM state. Afterwards, Fusion will begin restore the VM back to the state it was in when the target snapshot was initially taken.

Note: If your system is cluttered with snapshots you no longer need, access the target VM's snapshot menu, and highlight the snapshot you wish to delete. Click the trash bin icon to remove the old snapshot. Of course, the machines will question your resolve with a pop-up. Click the *Delete* button to proceed.



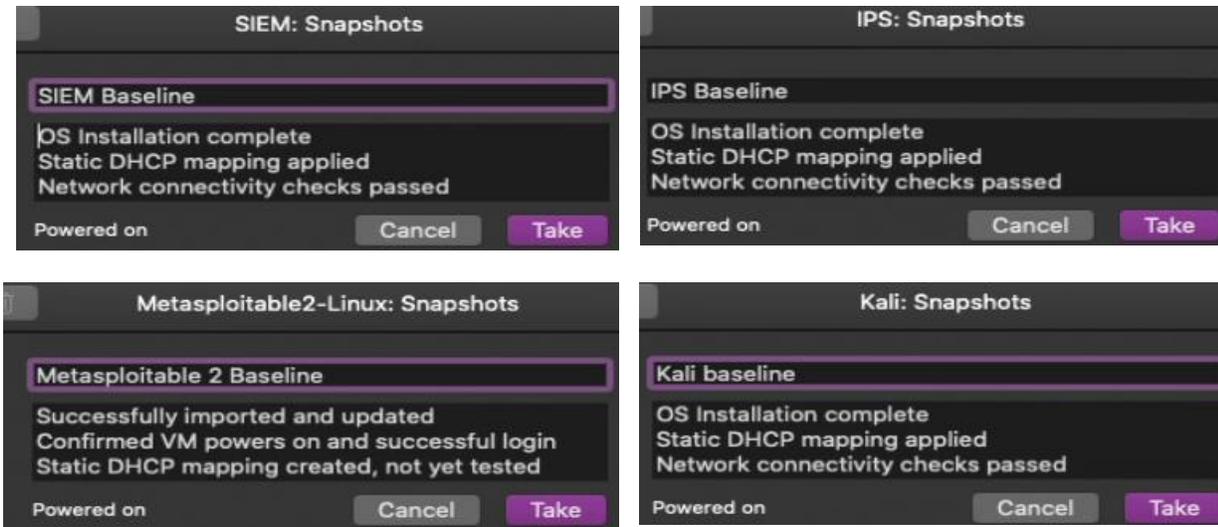
11-77: To delete a snapshot, navigate to the snapshots menu, highlight the target snapshot, click the trash bin icon, and confirm the action in the pop-up that appears by clicking the *Delete* button.



11-78: To restore a snapshot, right-click on the target VM in the *Virtual Machine Library* window and select *Snapshots*. In the *[Virtual Machine Name]: Snapshots* window, left click the target snapshot to highlight it, then click the restore button (square icon with a circular arrow) in the upper left corner. A pop-up will appear, asking if the user wants to save the VM in its current state before restoring (*Save* or *Don't Save*). Make a decision, and Fusion will begin restoring the target snapshot.

11.6.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines

Now that students understand how to create snapshots, it is highly recommended that they create baseline snapshots for the remaining virtual machines in their lab environment – the SIEM, IPS, kali, and metasploitable 2 virtual machines. In the chapters to come, there will be a lot of complicated configuration tasks that students will need to perform in order to enable different functionality for their environment. Having a baseline snapshot to fall back to in case there are problems completing a task is handy for troubleshooting purposes.



11-79: Now that students know how to create snapshots, apply that knowledge and make baseline snapshots for the other lab virtual machines. Having a baseline to fall back to in case something fails in the later chapters of this book is very important and will save students from a lot of headaches.

11.7 Chapter Review

Students should have all 5 virtual machines created for the baseline lab environment, as well as baseline checkpoints for all 5 virtual machines. It was a long journey to get to this point, but it's far from over. Here is a checklist of tasks to complete:

- Complete chapter 15, *Routing and Remote Access for Hosted Hypervisors*, starting on p. 727. In this chapter, students will learn how to enable SSH access to their lab virtual machines from Windows, Linux or MacOS. This functionality is vital for finishing the IPS and Splunk setup guides more easily than through the VM console alone.
- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in chapter 17, *Network Intrusion Detection*, starting on p. 980.
- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in chapter 18, *Setting up Splunk*, starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out chapter 19, *End of the Beginning*, starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out chapter 20, *Extra Credit*, starting on p. 1055.

Chapter 12 Patch Notes

- As a whole, this chapter is much more detailed than the VMware Workstation chapter in the first edition.
- Readers are informed about how to acquire Trial copies of VMware Workstation pro, as well as the VMware academic program for getting a discounted permanent license of the software.
- Detailed installation instructions have been made available for both Linux and Windows users.
- The Linux users are informed on how a lot of the Preferences functions require running vmware as the root user, and how access to the network manager will require sudo privileges. Windows users are informed that UAC approval will be required to make significant network changes.
- There's a sidebar discussion that discusses the performance advantages of allocating a larger chunk of RAM to VMware Workstation can benefit performance on systems with huge amounts of RAM, or how workstation can be forced to swap to disk more often if memory is tight on the host system
- Detailed instructions on how to configure an IP address for the VMnet1 adapter on both Windows and Linux, as well as the caveat that for reasons entirely beyond me, Linux still cannot persist hypervisor virtual interfaces, or maintain their network configurations when the system is rebooted.
- Apparently the Linux ip command can be used to assign multiple IP addresses to a single interface. This could pose problems for Linux workstations. The flush option has been added as an instruction to remove active IP address configurations on the vmnet1 interface before reconfiguring it for use in the lab environment.
- The Easy Install "feature" is remarkably annoying, especially when you're trying to make virtual machines in which you do not want vmware tools installed on. Students are guided on how to create their virtual machines to work around Easy Install, and advised to not install VMware Tools.
- As with the other chapters, a template file for doing asset management of their lab environment is provided to students.
- Detailed instruction is provided to both Linux and Windows users for copying the decompressed Metasploitable2-Linux directory to the default virtual machine directory to easily import the virtual machine, and ensure that the VM gets ran from the correct location, etc.
- To that effect, students are guided through the process of upgrading the virtual hardware of the metasploitable 2 VM.

Chapter 12: VMware Workstation Pro

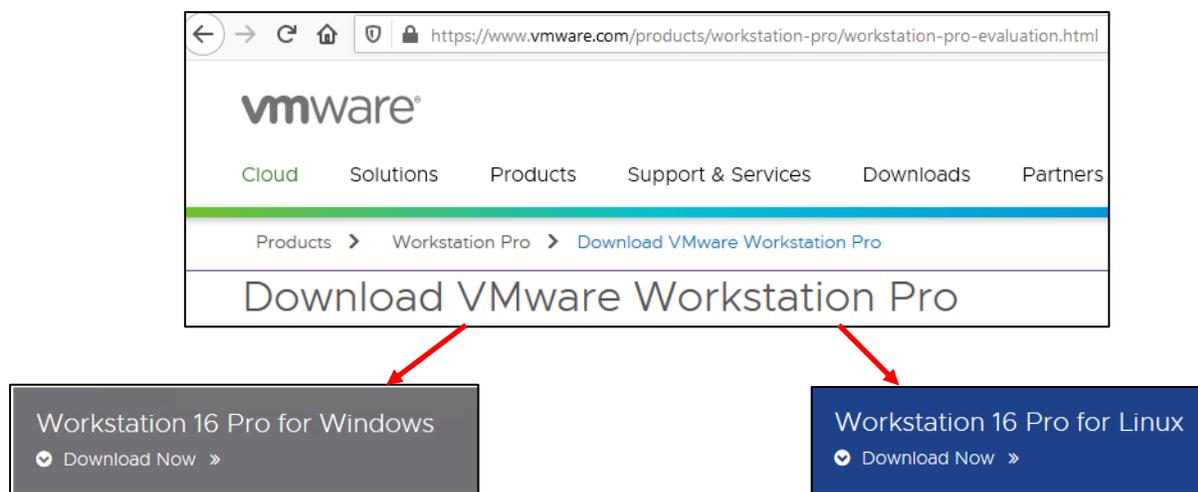
VMware Workstation Pro is a hosted hypervisor available for Windows and Linux. Like the other VMware hypervisors, it's been around for a very long time. This chapter will cover Installing and configuring Workstation Pro for both Windows and Linux hosts.

12.1 Installation

In the subsections that follow, students will learn how to install Workstation Pro on both Windows and Linux. Before doing that however, students will need to download the installer for their host operating system of choice. Visit the following link:

<http://www.vmware.com/products/workstation/workstation-evaluation.html>

And download the installer for Windows or Linux.



12-1: As they say, the journey of a thousand miles begins with a single step. Our first step is downloading the VMware Workstation Pro package for either Windows or Linux, depending on the host operating system students wish to use. Check out sections 12.1.1 or 12.1.2 for Windows or Linux installation instructions, respectively.

Getting the most Virtualization for your Money

Bear in mind that VMware Workstation Pro is not a free product. The link I provided above is for a 30-day free trial. After that, VMware will demand that you pay for a license. Sometimes, if you're lucky, your workplace will either provide a discount or reimburse you for the cost of the software outright.

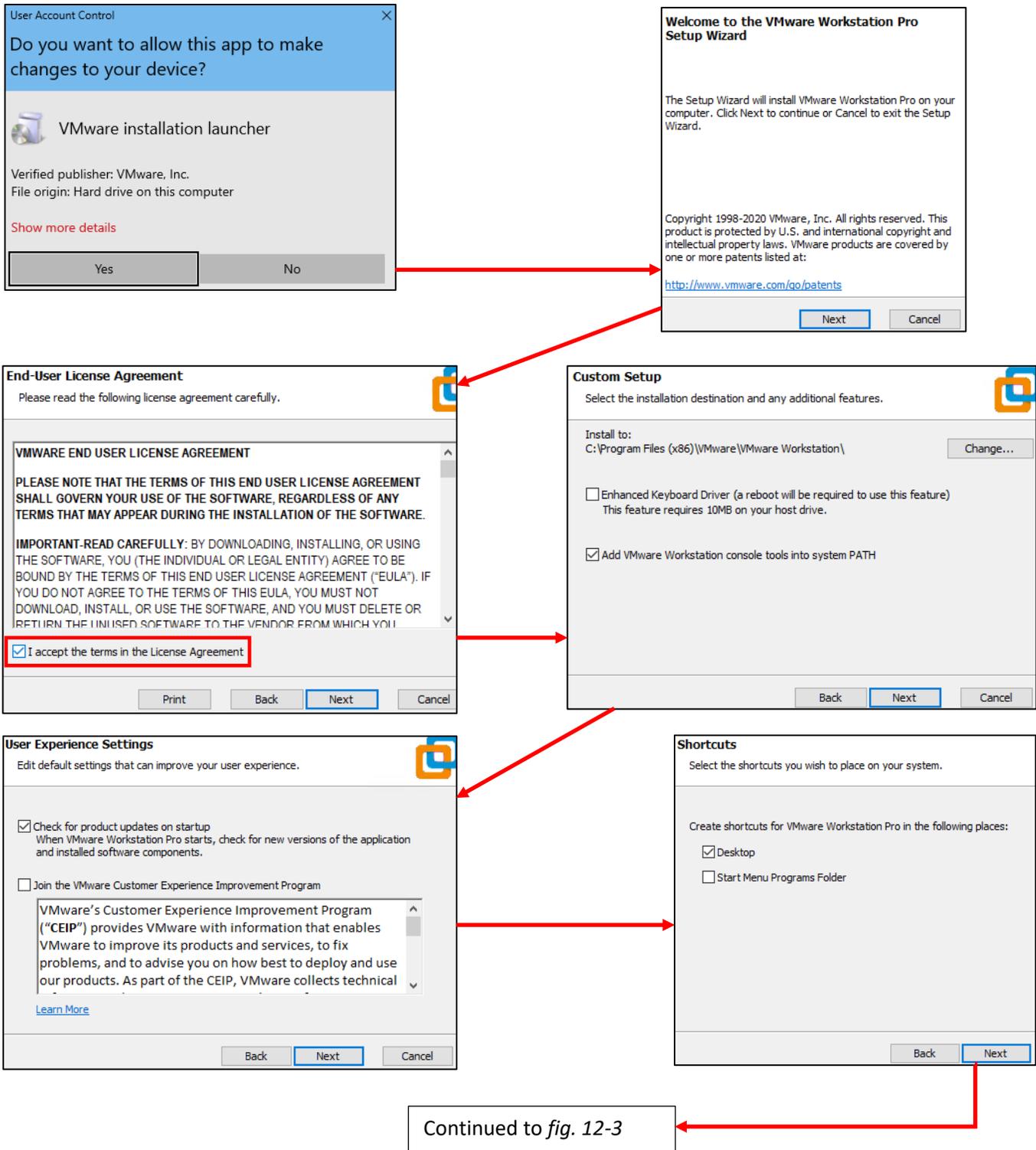
So, what should the rest of us do, who have to pay out of pocket? The best recommendation I can offer is to pay attention during the holidays, specifically "Black Friday" and "Cyber Monday". VMware will occasionally sell licenses for their products at a discounted rate. Additionally, they also provide a steep discount to users who sign up to vmware.com with a ".edu" e-mail address. Through the "VMware Academic Program".

12.1.1 Windows Installation Guide

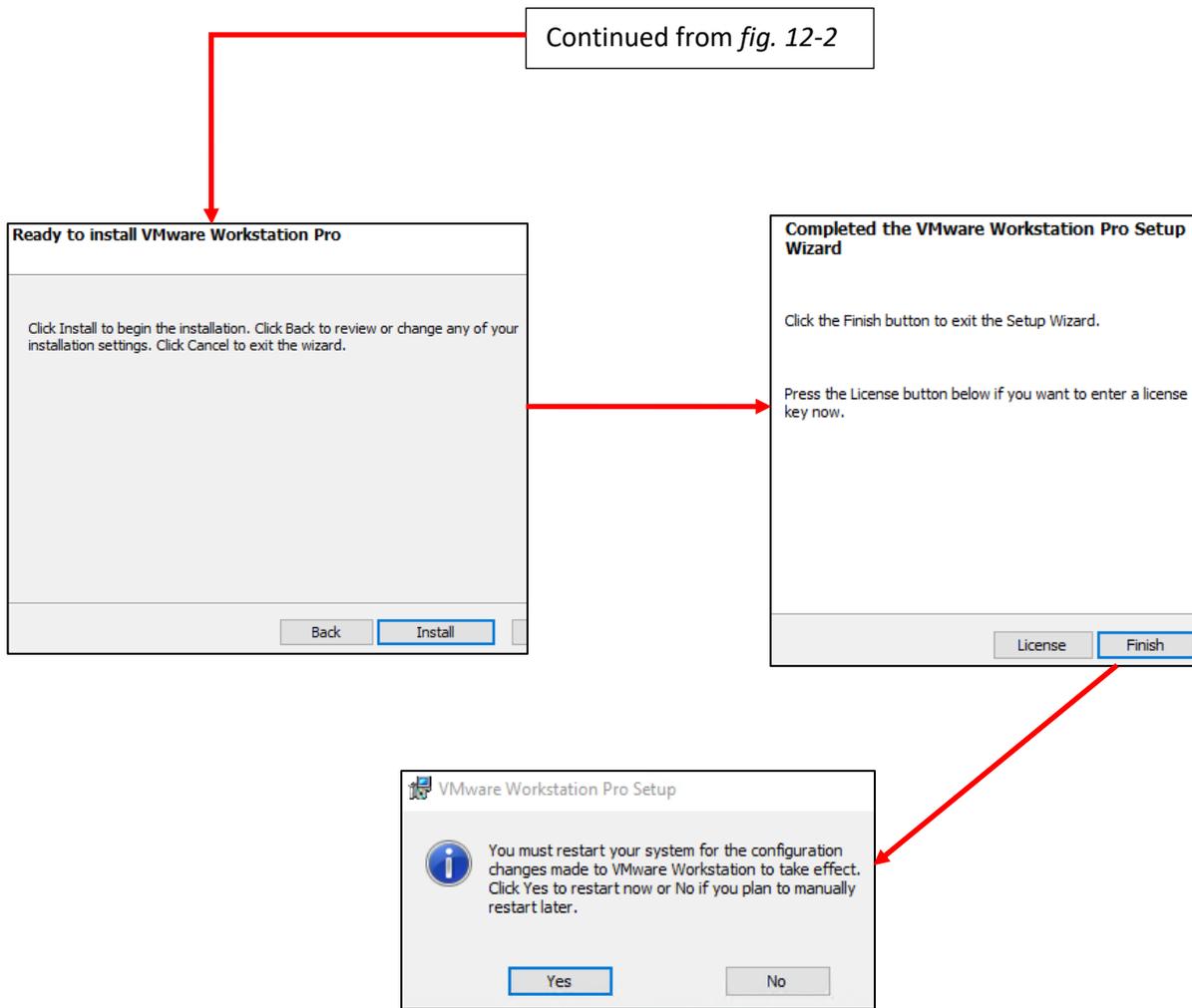
Once you've downloaded the Installer file, double click on it to begin. The first screen students should see is a UAC prompt (if it is enabled on your system) asking if they want to allow the installer to make changes to their system. Click *Yes* to continue, and a Window labeled *VMware Workstation Pro Setup* will appear. As always, there's nothing to do on the first screen, except click *Next* to continue. The next screen, titled End-User License Agreement requires users to accept VMware's licensing terms to continue. Click the checkbox labeled *I accept the terms of the License Agreement* (without reading them, as is tradition) and click *Next* to proceed.

The next screen is labeled *Custom Setup*, and if necessary, students can click the *Change* button in order to modify where on the system VMware Workstation will be installed. There are also two checkboxes referring to an enhanced keyboard driver, and adding console tools to the system PATH. There should be no need to modify these checkboxes, so click *Next* to accept the default settings. On the *User Experience Settings* screen, students can control whether or not Workstation will check for new updates upon starting up, and whether or not the system will participate in the Customer Experience Improvement Program. In a nutshell, VMware Workstation can send usage and telemetry data back to VMware for the goal of product improvement. Students are free to check or uncheck these options as they see fit, but it is recommended to check the *Check for product updates on startup* checkbox at a minimum. When finished, click *Next* to move on.

The next screen, labeled *Shortcuts* defines whether or not users want to create install shortcuts on their desktop, or in the start menu. When ready, click *Next* to proceed. Finally, a screen labeled *Ready to install VMware Workstation Pro* appears. Click the *Install* button to allow the wizard to continue. After a moment or two, the installation wizard will complete its task, and a screen labeled *Completed the VMware Workstation Pro Setup Wizard* will appear. Click *Finish* to exit the installer. Optionally, if students purchased a license key for VMWare Workstation Pro, they may click the *License* button to apply it immediately. Please be aware that after finishing the installer, your system will need to be rebooted. Click *Yes* to allow the installer to reboot Windows.



12-2: The installation lists the steps of the of Workstation Pro installer for Windows. For the most part, there are very few configuration options that students should need to change. There's a good chance that if there's a setting that does to be changed, students will know their environment better than I would. Otherwise, most can click *Next* to proceed through the installation wizard with no problems.



12-3: When students are done customizing their installation (or repeatedly mashing *Next*, I won't judge), click the *Install* button to let the installation wizard do all the heavy lifting. After a moment or two, a screen will appear, stating the installation is completed. If students purchased a VMware workstation pro License, they may click the *License* button to enter it now. Otherwise, click the *Finish* button, then reboot your system for the necessary configuration changes to take effect.

12.1.2 Linux Installation Guide

The process for installing VMware Workstation on Linux will be demonstrated using Ubuntu Desktop 20.04. The instructions provided assume readers already have either `root` or `sudo` access on their system to run the installer they have already downloaded, and have already installed Linux kernel headers for their distribution of choice (see chapter 1, [section 1.7](#), pp. 28-32). Additionally, VMware Workstation requires the `gcc` (GNU C Compiler) software package, so students will need to acquire that through their distribution's package manager as well (e.g., `sudo dnf install gcc elfutils-libelf-devel` for RHEL-based distributions, or `sudo apt-get install gcc` for Debian-based distros).

VMware Workstation on Linux is distributed as a `.bundle` file. Open a terminal window and navigate to the directory the `.bundle` file was downloaded to (usually, this is the current user's downloads directory, so in most cases students can run the command: `cd ~/Downloads`, followed by the `ls` command to confirm the `.bundle` file is present). The installer needs to be ran with root permissions. This can be done by running the command:

```
sudo bash VMware-Workstation-[Version Number]-[Build Release].x86_64.bundle
```

The name of the file is quite long, but students can try to use tab completion to fill out the name of the file quickly (tab completion is a feature in most modern linux shells that while typing out the name of a long file in a terminal session, if the tab key is pressed, the shell will try to guess what file the user is attempting to access). The command above runs the bash command interpreter with root access to execute the contents of the `.bundle` file. There are numerous other ways to run the installer as root, but this is the fastest and easiest method.

Once the installer is finished, the `vmware` executable is located at `/usr/bin/vmware`. Most of the time, `/usr/bin` will be a part of the command shell's `PATH` variable. This means, `virtualbox` can usually be ran with the command:

```
vmware &
```

If you get the error `command not found`, try:

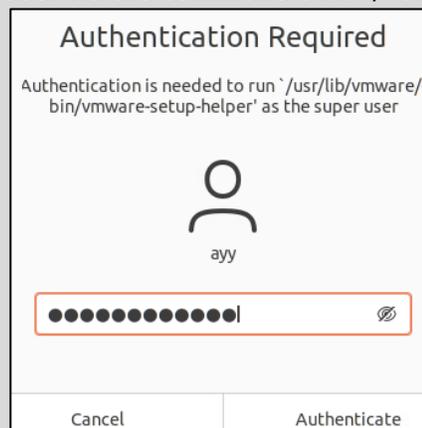
```
/usr/bin/vmware &
```

The ampersand (&) is a part of the command. It allows the user to keep using the terminal window while the `vmware` command is running. Be aware that if the terminal window used to run `vmware` is closed, the hypervisor will exit. If your distribution's window manager allows you to run VMware Workstation directly, or allows users to create a shortcut on a dock or desktop of some sort, it is recommended to use these methods to run the hypervisor, in order to prevent it from shutting down accidentally.

Upon running VMware Workstation for the first time, a window titled *Welcome to VMware Workstation* appears. Students are forced to accept two end-user license agreements to continue (one for `vmware workstation`, the other for a software component called OVF tool). Click the radio button labeled *I accept the terms in the license agreement* (without reading them, as is tradition) and click the *Next* button to proceed.

The next screen asks if students want to check for updates on startup. Select *Yes*, then click *Next*. The screen that follows asks if students would like to participate in the customer experience improvement program (CEIP) – in a nutshell, if you select *yes*, you agree to send usage and telemetry data in order to improve the product. After making a decision, click the *Next* button to proceed. Finally, users are given a choice to enter a license key (or follow a link to purchase a license key), or select the option to try VMware Workstation 16 for 30 days. If students have a license key, and they would like to enter it now, they may do so, otherwise, click the *I want to try VMware Workstation 16 for 30 days* radio button, and click *Finish* to close this initial setup wizard.

Note: When you finish running the initial setup wizard, you might see a pop-up asking you to enter your password so that the `vmware-setup-helper` function can use `sudo` to run with root privileges. Enter your password, then click the *Authenticate* button to continue. Alternatively, try running `vmware` as the root user via the command `sudo /usr/bin/vmware &`.



12-4: This pop-up is the Linux equivalent of a Windows UAC prompt, asking for permissions to make changes. Make sure your user has `sudo` access (usually enabled by default on Redhat/CentOS, and Ubuntu), enter your password, then click *Authenticate* to proceed. If you continue to run into authorization/permission problems, try running VMware Workstation as the root user, via the command: `sudo /usr/bin/vmware &`

```
ayy@ayy:~$ sudo apt-get install gcc
[sudo] password for ayy:
ayy@ayy:~$ cd ~/Downloads/
ayy@ayy:~/Downloads$ ls
VMware-Workstation-Full-16.0.0-16894299.x86_64.bundle
ayy@ayy:~/Downloads$ sudo bash VMware-Workstation-Full-16.0.0-16894299.x86_64.b
undle
Extracting VMware Installer...done.
Installing VMware Workstation 16.0.0
Configuring...
[#####] 100%
Installation was successful.
ayy@ayy:~/Downloads$ vmware &
```

VMware Workstation - End User License Agreement

I accept the terms in the license agreement.
 I do not accept the terms in the license agreement

Next

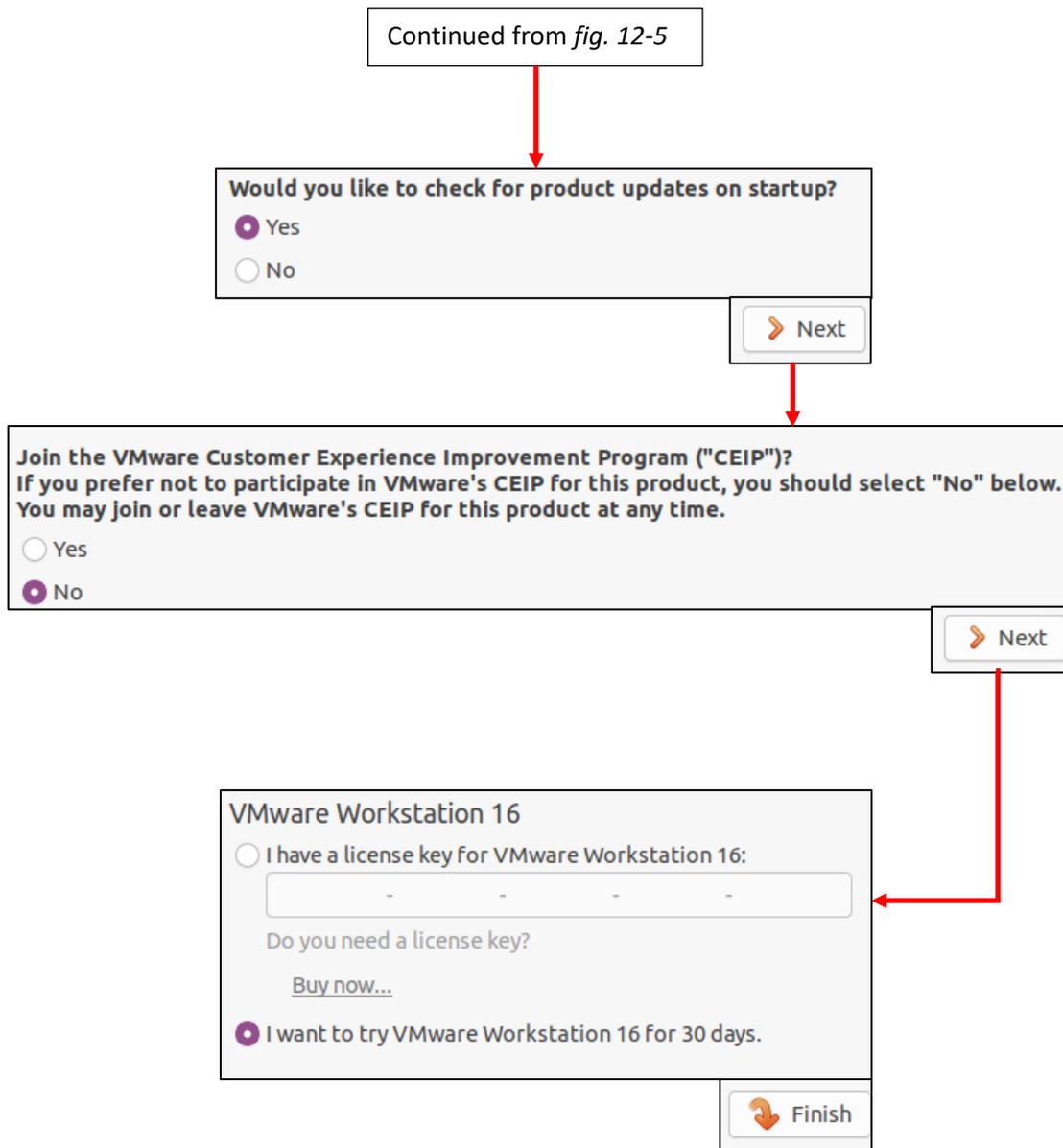
VMware OVF Tool component for Linux - End User License Agreement

I accept the terms in the license agreement.
 I do not accept the terms in the license agreement

Next

Continued to fig. 12-6

12-5: Assuming students have already downloaded the VMware Workstation installer bundle and Linux kernel headers, use the distro's package manager to acquire gcc, the GNU C Compiler software package. Afterwards, change directories to the location of installer bundle, and run it to begin the installation process. Once the installer completes, start VMware Workstation. Students will be required to answer a series of questions that are a part of an initial setup wizard. Accept the terms of the license agreement for both VMware Workstation, and the VMware OVF Tool to continue.



12-6: Next up, the setup wizard asks if users want to check for software updates on startup. Students are recommended to say *Yes*. The VMware CEIP is a mechanism for VMware to collect usage data and feedback from the users. Select *Yes* or *No* to continue, and finally, the setup wizard requests a license key for the software. If students purchased (or wish to purchase a license right now by clicking the *Buy now* link) they may enter a license key now. Otherwise, select the *I want to try VMware Workstation ## for 30 days* to continue using the free trial license, then click *Finish* to load the VMware Workstation interface.

12.2 Customizing VMware Workstation

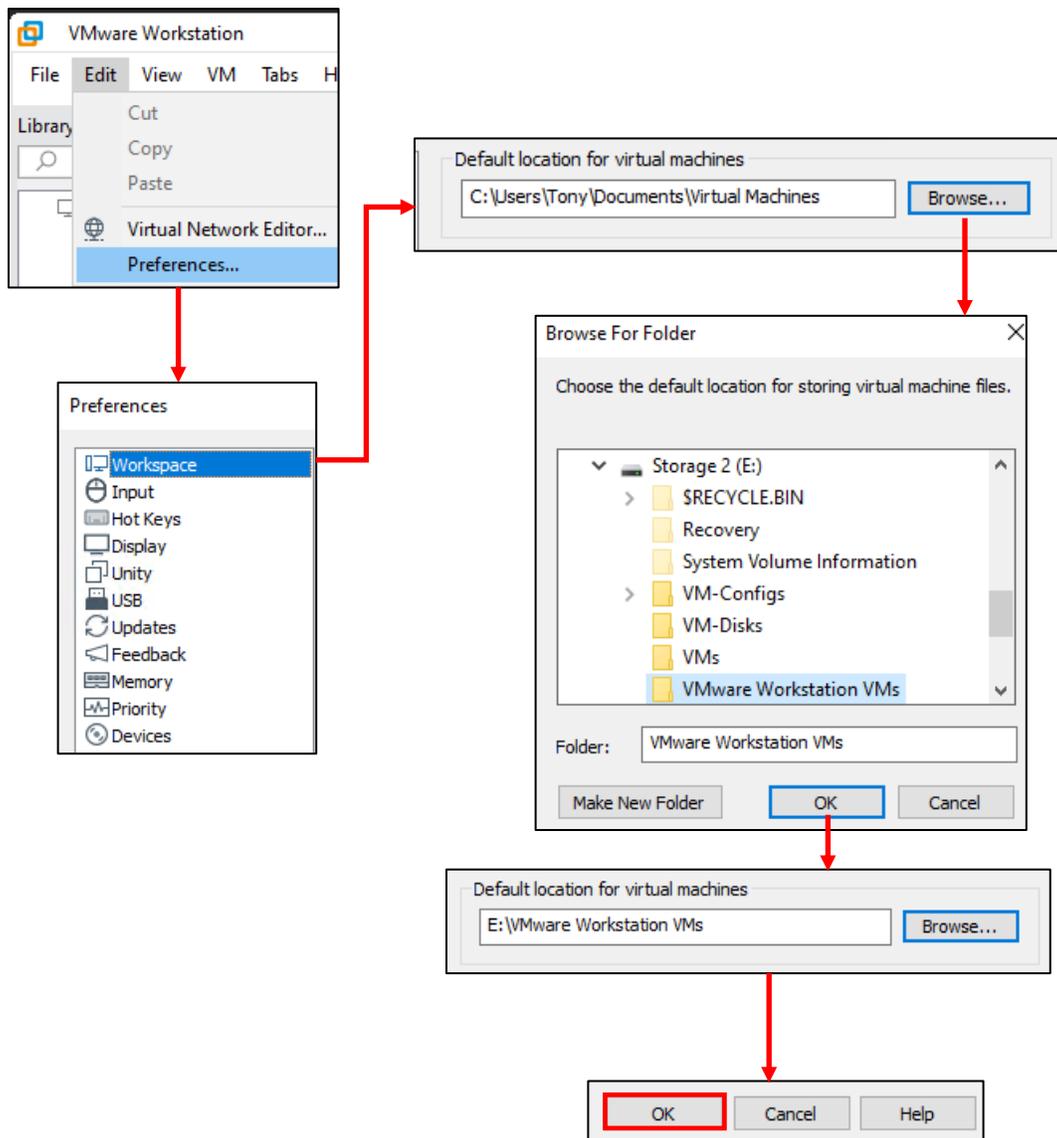
When starting VMware Workstation for the first time, users are greeted with welcome screen. Workstation has a host of customization options available that can be accessed by clicking *Edit* in the navigation menu followed by *Preferences*. The various configuration options can be accessed by clicking the icon and text on the left window pane with the setting users wish to modify. Many of the configuration options are pretty self-explanatory, and can be left with their default settings. However, there is at least one setting that students may wish to modify:

The *Workspace* option that has a small icon that looks like a PC next it. This option controls some of the general settings for VMware Workstation. Many of these settings are again, pretty self-explanatory and shouldn't require modification. Among these options however, is an input box labeled *Default location for virtual machines*. By default, VMware Workstation places virtual machine files under a folder named "Virtual Machines" on Windows, and "vmware" on Linux. This folder is located in the current user's home directory.

For Windows users, this is usually: C:\Users\[username]\Documents\Virtual Machines

For Linux users: /home/[username]/vmware

Replace the [username] placeholder above with the username of your account. Students may wish to change this setting to another directory on a separate, dedicated disk drive for better I/O performance. For example, on my Windows computer, I installed an SSD, and assigned it the E drive (E:), so I created a folder on that drive (E:\VMware Workstation VMs), clicked the *Browse* button, and selected that folder as the default location I want future virtual machines to be stored. Be sure to click the *OK* button in the bottom right corner of the *Preferences* window to apply any changes made, and close the window.



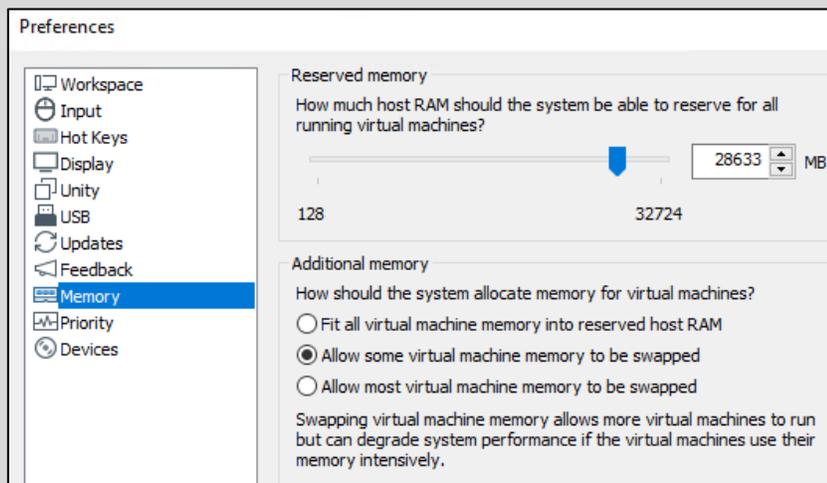
12-7: The *Preferences* menu contains a host of options that govern how VMware Workstation operates. Most of the options are pretty self-explanatory, with a few exceptions. The *Workspace* menu option contains a section labeled *Default location for virtual machines*. If students have dedicated storage they want to use for holding their virtual machines, create a new folder on that drive (or Linux partition) and click *Browse* to navigate to the newly created folder, select it, and click *OK*. To confirm this change, click the *OK* button in the bottom right corner of the *Preferences* menu to confirm any changes made and exit the menu.

Got it Memorized?

Are you running VMware workstation on a system with a lot of RAM? If you are, take a look at the *Memory* setting in the *Preferences* window. This option allows you to manually specify how much memory is reserved on the host for virtual machines, and also allows users to define whether or not they would like to allow virtual machine memory to be swapped to disk, and how frequently they would like that to happen. Recall in Chapter 5 where we discussed at length how the amount of memory and how fast your system's disk directly impacts how many virtual machines a physical host can support, as well as their performance. We discussed swapping to disk as well, and how it should generally be avoided.

If you have a system with 20GB of RAM (or more) I would recommend setting reserved memory allocation to 16GB (16,384MB). This ensures there is enough memory for all of the baseline VMs, and perhaps one or two more small virtual machines for those wishing to expand their lab environment a little bit. This also ensures that there is at least 4GB of RAM available for the host operating system.

On the other hand, if you're trying to fit all of the lab VMs on to a system that is barely meeting the minimum recommended requirements (even with trimmed down memory allocations), it might not be a bad idea to tell VMware Workstation to swap virtual machine memory to disk more often. This may cause poor VM performance, but will ensure the host operating system has access to RAM when it needs it for improved system stability.



12-8: The *Memory* option in the *Preferences* menu allows students to specify how much RAM to reserve from the host system for virtual machine use, as well how much and how often virtual machine memory should be swapped to disk. For students that are barely meeting the minimum requirements for the lab environments, selecting the option *Allow most virtual machine memory to be swapped* may free up more memory for the host operating system and improve host stability.

One last thing – If you're running VMware Workstation on Linux and attempting to change the *Memory* settings, you might have noticed this message:



You must be running Workstation as root to change these preferences.

12-9: This message will show up on the *Memory* options of the *Preferences* menu for users running VMware Workstation on Linux as a non-root user. You'll need to exit VMware workstation, and restart it with `sudo`, using the command below.

Exit VMware Workstation, and re-open it using the `sudo` command to give it root permissions:

```
sudo /usr/bin/vmware
```

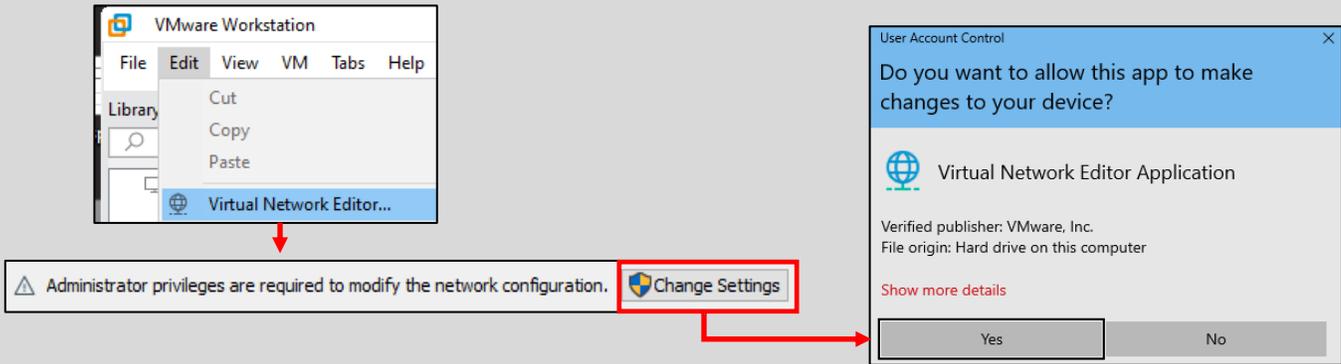
Open up the *Preferences* menu again, navigate back to *Memory* and you should be able to modify memory settings.

12.3 Virtual Network Editor

The Virtual Network Editor is, in a nutshell, the only reason student need VMware Workstation Pro over the standard edition. The network editor allows students to modify the configuration settings of the default virtual networks that ship with VMware Workstation, and create additional, custom virtual network segments. Students may access the network editor through the navigation menu. Click *Edit*, followed by *Virtual Network Editor*.

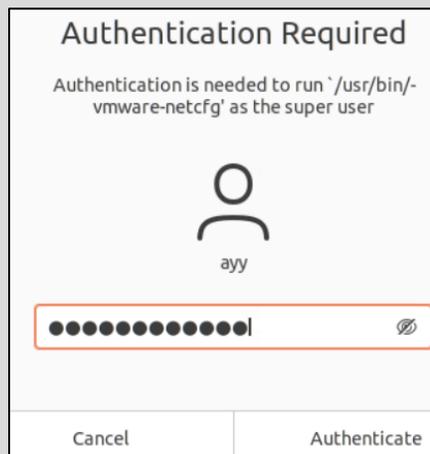
Note: Both Windows and Linux users will need special permissions to access or make changes in the Virtual Network Editor.

Windows Users: Upon loading the *Virtual Network Editor* window, towards the bottom of the menu there is a button labeled *Change Settings*. Click the button and a UAC prompt will appear requesting permission to make changes to the system. Click *Yes* to continue and users will be able to modify VMware Workstation's network settings.



12-10: Windows users will need to click the *Change Settings* button, then click *Yes* to the UAC prompt in order to make changes in the *Virtual Network Editor* menu.

Linux Users: If you are running VMware Workstation as a non-root user, a prompt very similar to the one that appeared before the initial setup wizard will appear asking the users to input their password to proceed. Enter your password, and if your user has `sudo` access, the *Virtual Network Editor* menu should appear.

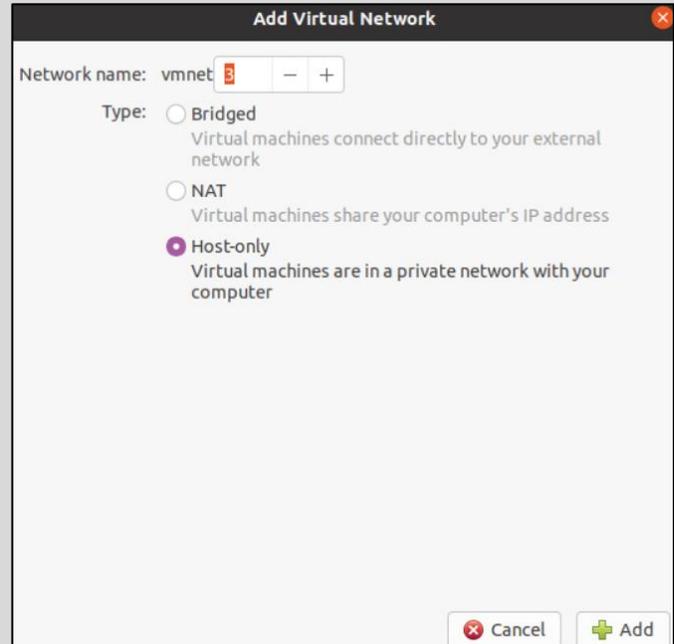
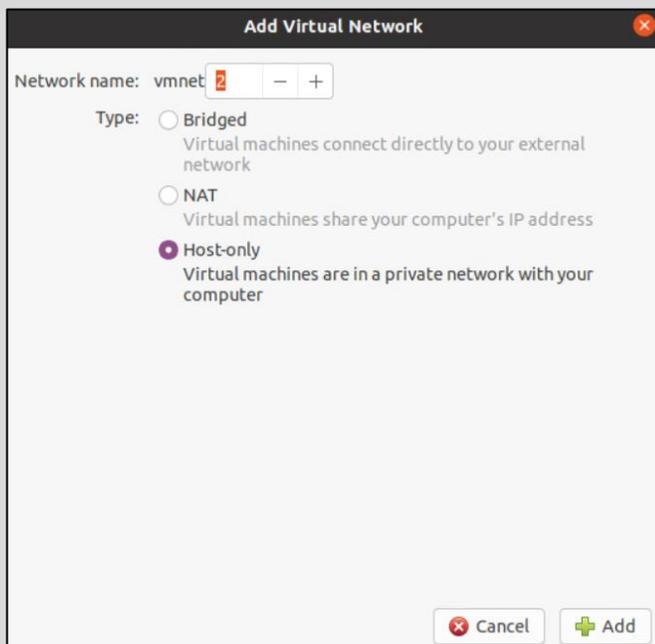


12-11: Linux users will be prompted with this pop-up again asking for their password. Enter your password, and if your user has `sudo` access, the *Virtual Network Editor* menu should appear.

By default, VMware Workstation ships with three virtual network segments defined – VMnet0, VMnet1, and VMnet8. VMnet0 serves as the bridged network segment, VMnet1 acts as a host-only network, and VMnet8 is a NAT network segment (if students need to review the different network segment types, check out chapter 4, [section 4.1](#), pp. 44-49). We'll need to create two more virtual network segments, and make some configuration changes while we're here. Perform the following actions using the *Virtual Network Editor* menu:

Click the *Add Network* button. A new window pops up titled, *Add a Virtual Network*. In the drop-down labeled *Select a network to add*, select VMnet2, then click *OK*. Repeat this process one more time, selecting VMnet3 from the drop-down. These two new network segments will serve as the IPS1 and IPS2 networks for the lab environment.

Note: The window that pops up for Linux users is slightly different. The title of the window is *Add Virtual Network*, and users are provided with an input box to specify the number network segment that Workstation will create. Additionally, students will be required to choose the *Type* of virtual network they wish to create. Be sure to select *Host-Only* for both vmnet2 and vmnet3.



12-12: The Window that pops up when users click *Add Network* on Linux is slightly different. Be sure to specify the *Type* as *Host-only* when creating vmnet2 and vmnet3.

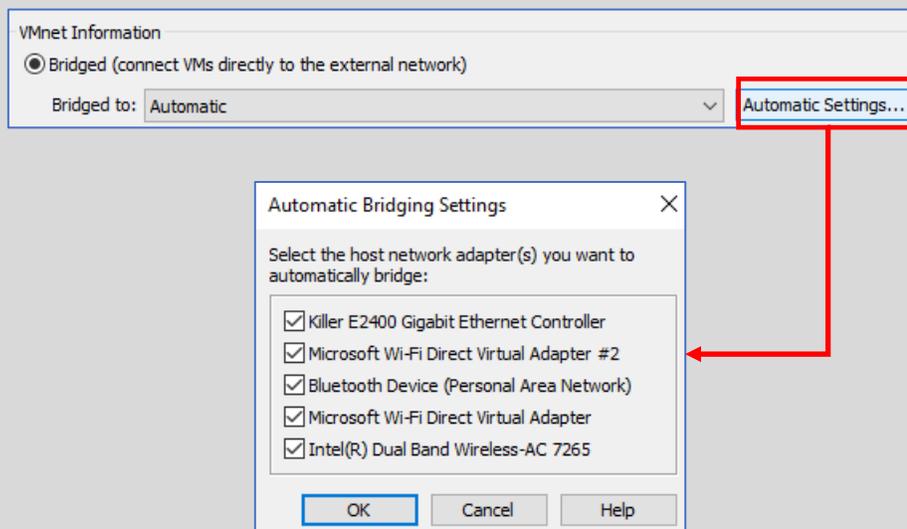
In the large window that contains a list of virtual network segments, click on the entry labeled VMnet0 to highlight it. Notice that the *Type* column for VMnet0 is set to *Bridged*, and the *External Connection* is set to *Auto-bridging*. This means that **by default, VMnet0 will attempt to bridge to any physical network interface on the host that has a network connection.**

In some cases, students may have a system with multiple network interfaces and this configuration setting allows for fault tolerance. If one network interface is unavailable for some reason, VMware Workstation will attempt to bridge using another network interface instead. In other cases however, students may want VMnet0 to bridge only to specific network interfaces.

Below the list of configured network segments are a variety of configuration options that can be applied to the currently highlighted virtual network segment (in this case, VMnet0). In the section labeled *VMnet Information*, there are three Radio buttons that defines the *Type* of virtual network segment the currently selected network will inherit. Currently the *Bridged (connect VMs directly to the external network)* radio button should be selected. Underneath this radio button is a drop-down labeled *Bridged to*. Currently this is set to *Automatic*. By clicking the drop-down, students can select a specific network interface to bridge VMnet0 to. For example, on my laptop, I want to bridge VMnet0 to wired network interface, *Killer E2400 Gigabit Ethernet Controller*. Click on the desired network interface to update the drop-down menu selection.

Rapid Holo Targeting

While working on VMnet0, you might have noticed the button to the right of the *Bridged to* drop-down labeled *Automatic Settings*. Click on the button, and a window appears that allows students to select (by checking or unchecking) which interfaces on the host that VMnet0 will attempt to bridge to when they are available, so long as the *Bridged to* drop-down is set to *Automatic*. This allows students to configure redundancy for the bridged network, or restrict which network interfaces VMnet0 is allowed to automatically bridge to.



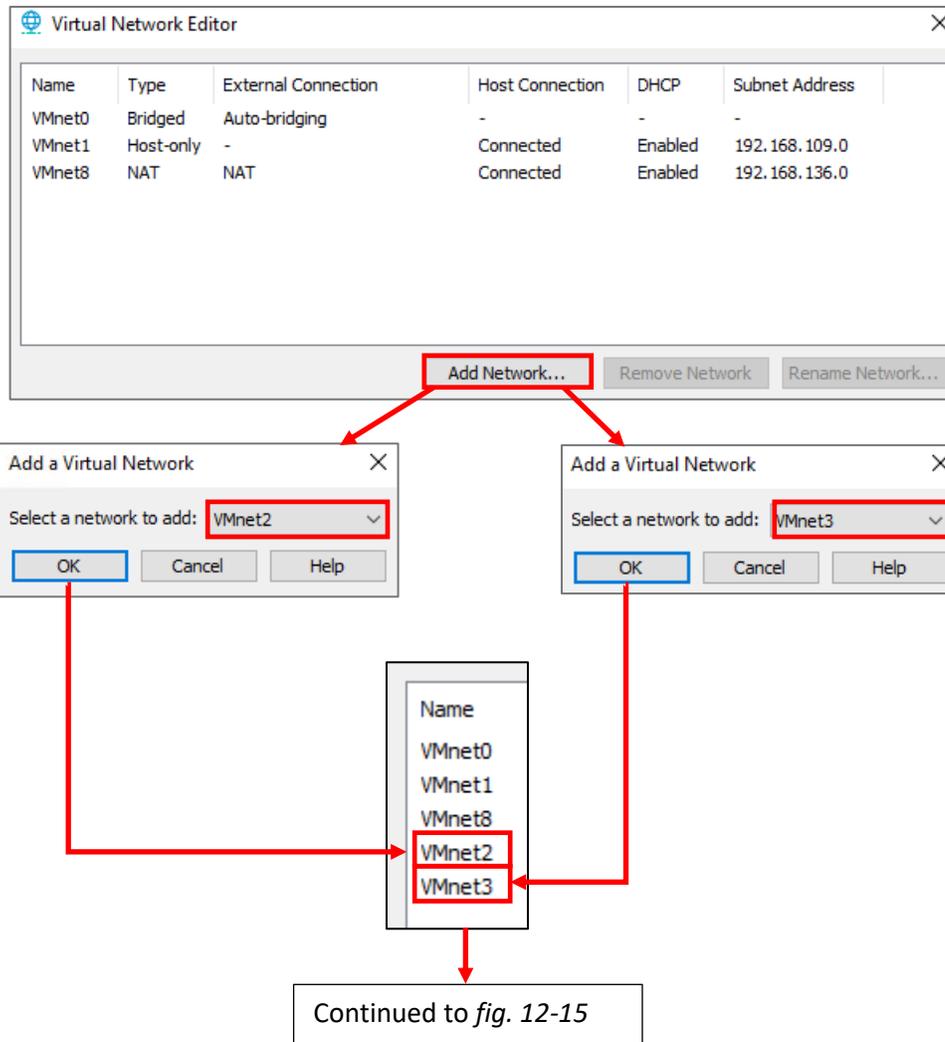
12-13: The *Automatic Settings* button allows users to define specific physical interfaces VMnet0 is allowed to bridge to. Use this if you need to restrict what physical interfaces VMware Workstation is allowed to bridge to.

Next, click on VMnet1. In the *VMnet Information* section, ensure the *Host-only (connect VMs internally in a private network)* radio button is selected. Immediately below this radio button, make sure that the *Connect a host virtual adapter to this network* checkbox is selected, and that

the *Use local DHCP service to distribute IP address to VMs* checkbox is unselected. **It is extremely important to disable the VMware DHCP service for this network segment.**

Click on VMnet2 to highlight it, ensure that the *Host-only* radio button is selected, then, uncheck **both** the *Connect a host virtual adapter to this network* checkbox **AND** the *Use local DHCP service to distribute IP address to VMs* checkbox. **Both of these options must be unchecked.** Repeat this process for VMnet3. **It is extremely important that both VMnet2 and VMnet3 both have the VMware DHCP service disabled and that there is never a host virtual adapter attached to either of these networks.**

When students are finished making these configuration changes, click the *Apply* button, then click *OK* to exit the *Virtual Network Editor*.



12-14: Welcome to the *Virtual Network Editor*. There are a few things students will need to do here before we can continue. To begin, students will need to create the VMnet2 and VMnet3 virtual networks. Click on the *Add Network* button. A pop-up appears labeled *Add a Virtual Network*. Select VMnet2 from the drop-down and click *OK*. Repeat this process again for VMnet3.

Continued from *fig. 12-14*

Name	Type	External Connection	Host Connection
VMnet0	Bridged	Killer E2400 Gigabit Ethernet...	-
VMnet1	Host-only	-	Connected
VMnet8	NAT	NAT	Connected
VMnet2	Host-only	-	Connected
VMnet3	Host-only	-	Connected

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Killer E2400 Gigabit Ethernet Controller

VMnet1 Host-only - Connected

VMnet8 NAT NAT Connected

VMnet2 Host-only - Connected

VMnet3 Host-only - Connected

Add Network...

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Automatic

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet1

Use local DHCP service to distribute IP address to VMs

VMnet2 Custom - -

VMnet3 Host-only - Connected

Add Network...

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Automatic

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet2

Use local DHCP service to distribute IP address to VMs

VMnet3 Custom - -

Add Network...

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Automatic

NAT (shared host's IP address with VMs)

Host-only (connect VMs internally in a private network)

Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet3

Use local DHCP service to distribute IP address to VMs

OK Cancel Apply

12-15: Students will need to make the following changes:

VMnet0: It is recommended to set the *Bridged to* drop-down to a specific network card. If students wish to bridge to more than one network adapter, check out the sidebar conversation, *Rapid Holo Targeting* (p. 419).

VMnet1: Verify the Host-only radio button is selected. Uncheck the *Use local DHCP service* checkbox

VMnet2: Verify the Host-only radio button is selected. *Uncheck the Connect a host virtual adapter, and Use local DHCP service* checkbox

VMnet3: Verify the Host-only radio button is selected. *Uncheck the Connect a host virtual adapter, and Use local DHCP service* checkbox

It is extremely important that the VMware DHCP service is disabled for VMnet1, 2 and 3 and that the host virtual adapter is disabled for VMnet2 and 3. Once finished Click *Apply*, then *OK* to close the *Virtual Network Editor*.

12.4 Configuring the VMnet1 Host Virtual Adapter

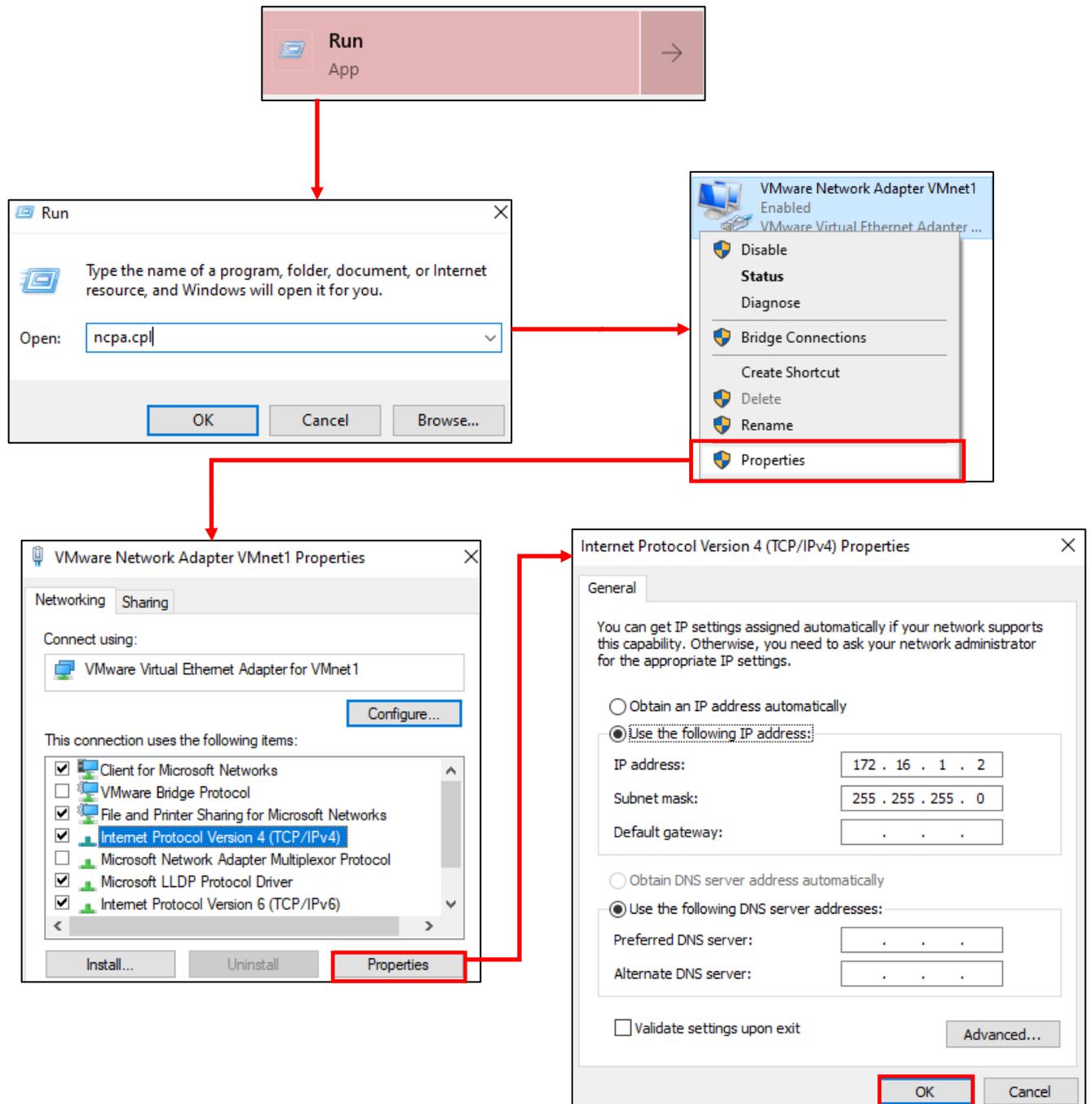
Before students can continue, they will need to configure the VMnet1 host virtual adapter with an IP address. This section will guide students on how to configure the vmnet1 host virtual adapter on Linux and Windows.

12.4.1 Configure the VMnet1 Host Virtual Adapter on Windows

Begin by opening the Windows *Run* prompt (Click the *Start* button and search for the *Run* application), type `ncpa.cpl` into the *Open* input box, then hit enter. This will open up the *Network Connections* panel without the need to figure out where in the UI Microsoft hid it in the latest Windows release.

Locate the network adapter icon titled *VMware Network Adapter VMnet1*. Right-click on it, and select *Properties*. This opens a new window titled *VMware Network Adapter VMnet1 Properties*. In the box labeled, *This connection uses the following items*, locate the entry labeled *Internet Protocol Version 4 (TCP/IPv4)*, left-click to highlight it, then click the *Properties* button.

This opens a new window labeled *Internet Protocol Version 4 (TCP/IPv4) Properties*. In the *General* tab below, make sure that the *Use the following IP address* radio button is selected, as well as the *Use the following DNS server addresses* radio button. In the *IP address* input box enter 172.16.1.2, and in the *Subnet mask* input box, enter 255.255.255.0. **All of the other remaining input boxes should be left blank.** Click *OK* to apply these settings.



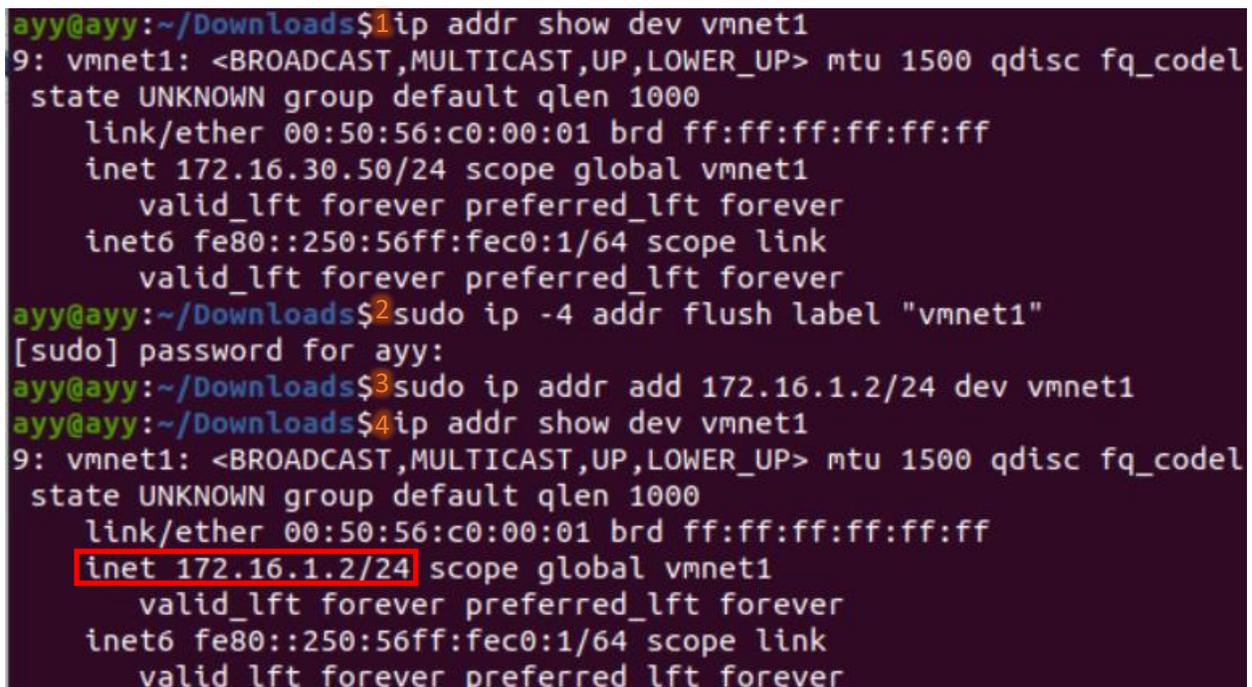
12-16: Open the Windows *Run* prompt, then type `ncpa.cpl` and hit enter. This opens up the *Network Connections* panel. Find the interface named *VMware Network Adapter VMnet1*, right click on it and select *Properties*. Click on *Internet Protocol Version 4 (TCP/IPv4)* in the *This connection uses the following items* window to highlight it, then click *Properties*. Click the *Use the following IP address,* and *Use the following DNS server addresses* radio buttons. Enter 172.16.1.2 as the IP address, and 255.255.255.0 as the Subnet mask. ***Leave all the remaining fields blank,*** then click *OK*.

12.4.2 Configuring the vmnet1 Host Virtual Adapter on Linux

Open a terminal application and run the following commands:

```
ip addr show dev vmnet1
sudo ip -4 addr flush label "vmnet1"
sudo ip addr add 172.16.1.2/24 dev vmnet1
ip addr show dev vmnet1
```

Students run the `ip` command four times. The first time is to verify `vmnet1` exists. The second time is to remove any pre-configured IPv4 addresses that may have been assigned to the `vmnet1` interface by VMware Workstation. Note that this command requires root permissions and is prefixed with `sudo`. The third command, also ran with `sudo`, is to manually set the IP address and subnet mask needed for `vmnet1`. The fourth and final command is to confirm the IP address and subnet was properly applied to the `vmnet1` interface. The `inet` field should contain the value `172.16.1.2/24`.



```
ayy@ayy:~/Downloads$1 ip addr show dev vmnet1
9: vmnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UNKNOWN group default qlen 1000
    link/ether 00:50:56:c0:00:01 brd ff:ff:ff:ff:ff:ff
    inet 172.16.30.50/24 scope global vmnet1
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fec0:1/64 scope link
        valid_lft forever preferred_lft forever
ayy@ayy:~/Downloads$2 sudo ip -4 addr flush label "vmnet1"
[sudo] password for ayy:
ayy@ayy:~/Downloads$3 sudo ip addr add 172.16.1.2/24 dev vmnet1
ayy@ayy:~/Downloads$4 ip addr show dev vmnet1
9: vmnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
state UNKNOWN group default qlen 1000
    link/ether 00:50:56:c0:00:01 brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.2/24 scope global vmnet1
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fec0:1/64 scope link
        valid_lft forever preferred_lft forever
```

12-17: The first `ip` command (1) is used to make sure `vmnet1` exists. The second `ip` command (2), ran with `sudo` privileges, is used to unset any IPv4 addresses that may have been configured for `vmnet1`. The third `ip` command (3) is also ran with `sudo` privileges to set the `vmnet1` IP address to `172.16.1.2` with a subnet mask of `255.255.255.0` (/24). The fourth and final `ip` command (4) is to confirm that the IP address of `vmnet1` is `172.16.1.2/24`.

Linux Students: `vmnet1` and its IP address settings have disappeared. Why?

Some of you may have noticed that upon rebooting their Linux host, `vmnet1` and any interface configurations performed (like, say setting the IP address and subnet mask) are gone. Starting VMware Workstation will at least bring back `vmnet1`, but the IP address configuration will still be gone. Unfortunately, this is something of a known issue with hosted hypervisors on Linux.

After having done this for many years, I am still unaware of any method of getting the `vmnet1` interface itself, and/or any customizations performed against the interface to persist between reboots on Linux. The only method that seems to work is restarting VMware Workstation, and reperforming the commands in section 12.4.2 to reset the IP address of the `vmnet1` interface every time the Linux host is rebooted.

Fortunately, I've created a simple script for Linux users that automates this process, called Flightcheck-Linux. We'll talk more about this script in Chapter 15. For now, I ask readers to bear in mind that every time the host system reboots, they will be required to restart VMware Workstation, and reconfigure the IP address of `vmnet1` manually.

12.5 Building the first Virtual Machine, pfSense

The pfSense virtual machine is responsible for binding the entire lab environment together. It is a well-supported firewall distribution with amazing ease of use and functionality. pfSense is also very modular, featuring a system for adding on additional functionality through BSD's `pkg` software package manager.

It is recommended for students to download all of the ISOs (e.g., pfSense, Ubuntu Server and Kali Linux), and pre-built virtual machines (e.g. Metasploitable 2) required for their lab environment in advance. Check out chapter 1, [section 1.5.4](#) (p. 26) for download links. Additionally, students must decompress the pfSense installation ISO before attempting to boot from it. [Section 1.8](#) (pp. 33-35) covers how to perform this task.

12.5.1 VM Creation

VMware Workstation virtual machines are created through a step-by-step process called the New Virtual Machine Wizard. Open the VMware Workstation application. On the central screen, students may click the *Create a New Virtual Machine* button, or select *File > New Virtual Machine* to start the wizard.

A new window will appear, welcoming students to the *New Virtual Machine Wizard*. The screen also prompts students to select the type of configuration they would like use. Click on the Radio button labeled *Typical (recommended)* and click *Next* to continue.

The next screen is titled *Guest Operating System Installation*. Click on the radio button labeled *Installer disc image file (iso)*, then click the *Browse* button to open your host operating system's file browser. Navigate to the directory where the decompressed pfSense ISO is located, and select it. Afterwards, click *Next* to proceed.

The next screen, labeled *Name the Virtual Machine* asks students to name the virtual machine and confirm where they would like to store the VM's files. Rather unhelpfully, the wizard defaults to naming the virtual machine "FreeBSD version 10 and earlier 64-bit" and sets that as the name of the directory to store the virtual machine's files in, under the *Default location for virtual machines* specified in the *Preferences* menu (see section 12.2). For example, if the Default virtual machine location is set to `E:\VMware workstation VMs`, the *Location* will be set to `E:\VMware Workstation VMs\FreeBSD version 10 and earlier 64-bit`. In the *Virtual machine name* input box, enter pfSense. The *Location* input box should automatically change to reflect the new name students give their virtual machine. Click the *Next* button to continue.

The *Specify Disk Capacity* screen appears. In the input box labeled *Maximum disk size (GB)*, change this setting to 5.0 (the virtual machine's disk will be 5GB in size). Underneath, click the radio button labeled *Store virtual disk as a single file*, then click *Next*.

The final screen, labeled *Ready to Create Virtual Machine* provides a brief description of the configuration settings VMware believes are acceptable defaults, alongside the configuration choices students have made through the wizard. As always, they are wrong. Click the *Customize Hardware* button. This brings up a new window labeled *Hardware*.

On the left-hand pane, under the column labeled *Device*, click on the *Memory* listing to highlight it. The right pane updates to display settings related to the amount of memory that will be allocated to this virtual machine. In the input box labeled *Memory for this virtual machine*, change this to 512 (that is, 512MB of RAM will be allocated to the pfSense VM). Next, click the *USB Controller* entry in the *Device* column to highlight it, then click the *Remove* button at the very bottom of the left pane. This will remove the USB controller hardware from the virtual machine. Repeat this process for the *Sound Card* device entry as well in order to remove it.

Next, click on *Network Adapter*. On the right pane, under the *Network connection* section, click the *Bridged: Connected directly to the physical network* radio button. Make sure that the checkbox *Connect at power on* is checked under the *Device status* section (This should already be selected by default). Click the *Advanced* button below the Network Connection section. A new window appears, labeled *Network Adapter Advanced Settings*. Near the bottom of this new window is a section labeled *MAC Address*, and a button labeled *Generate*. Click the button, and a series of letters and numbers will appear in the input box. This is a MAC address. Copy the contents of this input box, and click *OK* to exit.

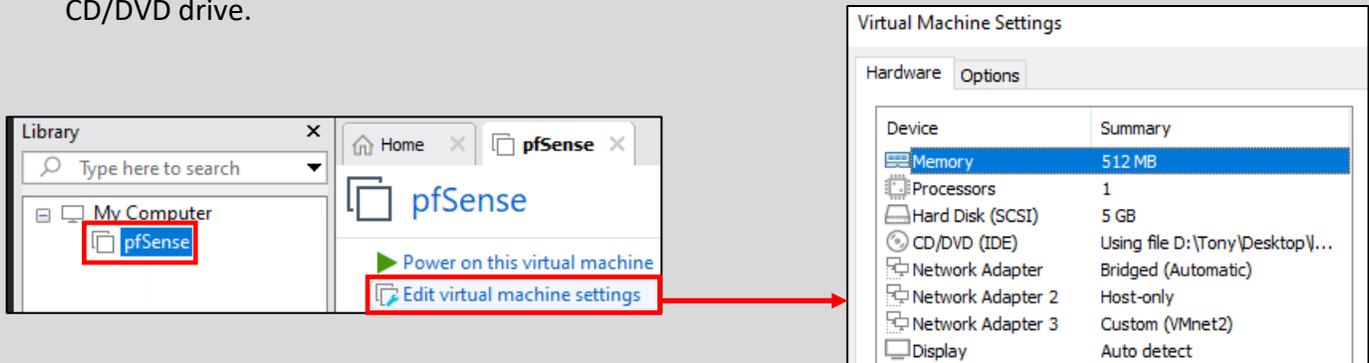
Next, click the *Add* button at the bottom of the left pane. The *Add Hardware Wizard* appears. In the pane labeled *Hardware types*, select *Network Adapter*, and click *Finish*. This will close the window, and the newly installed hardware, labeled *Network Adapter 2*, will be highlighted in the Hardware window. Under the Network connection section, select the radio button labeled *Host-only: A private network shared with the host*, and under *Device status*, ensure that the *Connect at power on* checkbox is selected.

Repeat this process once more and add another *Network Adapter* through the *Add Hardware Wizard*. Back in the *Hardware* window, with *Network Adapter 3* highlighted, under the Network connection section, click the radio button labeled *Custom: Specific virtual network*, and in the drop-down below the radio button, select *VMnet2* (**Note:** Linux users will select */dev/vmnet2* instead). Finally, ensure the *Connect at power on* checkbox under *Device status* is checked as well.

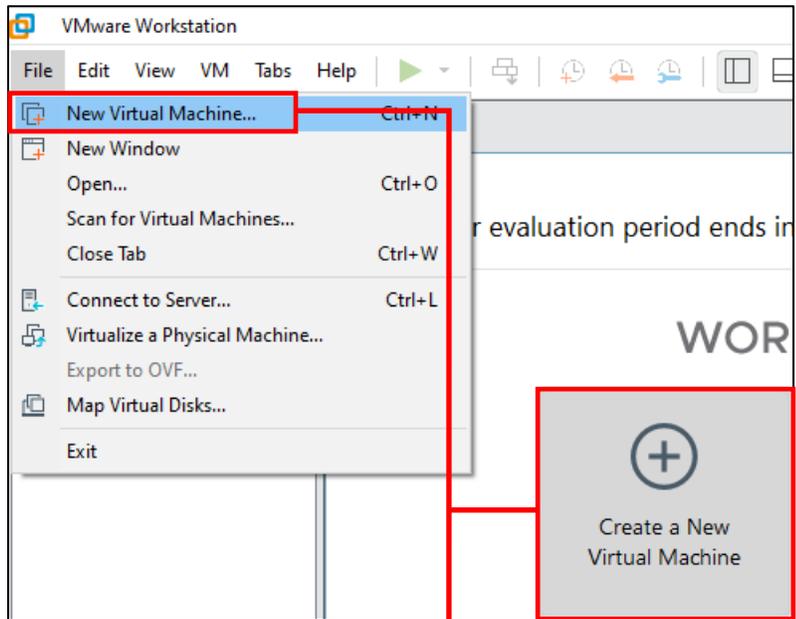
Once students have finished performing all of these tasks, click the *Close* button in the bottom right corner of the *Hardware* window to exit back to the New Virtual Machine Wizard's *Ready to Create Virtual Machine* window. The primary pane will update to reflect the changes made to the pfSense virtual machine. uncheck the *Power on this virtual machine after creation* checkbox, then click the *Finish* button to complete the wizard, and create the pfSense VM.

Jumped The Gun

If you accidentally clicked *Finish* on the *Ready to Create Virtual Machine* window without making the hardware changes above, no worries! In the main VMware Workstation window, on the left pane labeled *Library*, left click on the pfSense VM entry to highlight it, then click on *Edit virtual machine settings* to open up a window labeled *Virtual Machine Settings*. You can make all of the changes above on the *Hardware* tab in this window. Keep this in mind for later – we'll be back here after we're done installing the pfSense operating system in order to remove the virtual CD/DVD drive.



12-18: Did you accidentally click *Finish* instead of the *Customize Hardware* button on the last screen of the *New Virtual Machine wizard*? No problem. Highlight the pfSense VM, and click *Edit virtual machine settings* to bring up a window that will allow you to customize the virtual hardware.



Continued to *fig. 12-20*

12-19: Access the New Virtual Machine Wizard by either click the *Create a New Virtual Machine* button on the home tab of the VMware Workstation main window, or by Clicking *File > New Virtual Machine* from the navigation menu. On the Welcome screen, click on the *Typical (recommended)* radio button, then click *Next* to proceed.

Continued from *fig. 12-19*

Guest Operating System Installation
A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

Installer disc:
No drives available

Installer disc image file (iso):
[Empty text box] **Browse...**
⇒ Select the installer disc image to continue.

I will install the operating system later.
The virtual machine will be created with a blank hard disk.

Browse for ISO Image

← → ↕ ↑ > This PC > Desktop > lab_isos >

Organize ▾ New folder

3D Objects	Name
C on BAHAMUT	Metasploitable2-Linux
D on BAHAMUT	kali-linux-2020.3-installer-amd64.iso
Desktop	pfSense-CE-2.4.5-RELEASE-p1-amd64.iso

Installer disc image file (iso):
D:\Tony\Desktop\lab_isos\pfSense-CE-2.4.5-RELEASE [Browse...]
[i] FreeBSD version 10 and earlier 64-bit detected.

< Back **Next >** Cancel

Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
pfSense|

Location:
E:\VMware Workstation VMs\pfSense [Browse...]
The default location can be changed at Edit > Preferences.

< Back **Next >** Cancel

Continued to *fig. 12-21*

12-20: On the *Guest Operating System Installation* screen, select the *Installer disc image file (iso)* radio button, then click the *Browse* button. Locate the decompressed pfSense ISO, select it then proceed to the *Name the Virtual Machine* screen. In the *Virtual machine name* input box, enter pfSense, then proceed by clicking *Next*.

Continued from *fig. 12-20*

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB):

Recommended size for FreeBSD version 10 and earlier 64-bit: 20 GB

Store virtual disk as a single file

< Back Next > Cancel

Ready to Create Virtual Machine
Click Finish to create the virtual machine and start installing FreeBSD version 10 and earlier 64-bit.

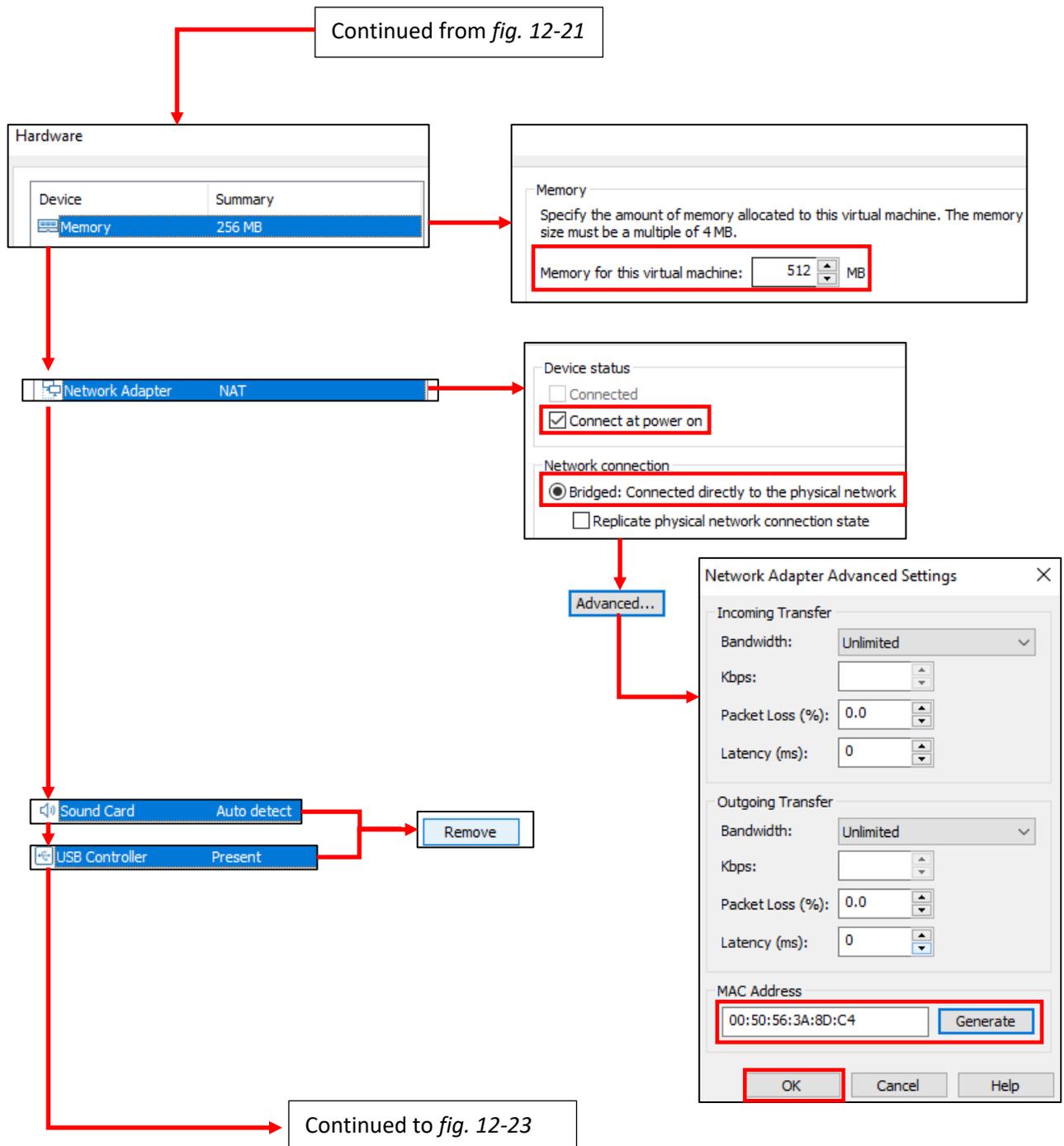
The virtual machine will be created with the following settings:

Location:	E:\VMware Workstation VMs\pfSense
Version:	Workstation 16.x
Operating System:	FreeBSD version 10 and earlier 64-bit
Hard Disk:	5 GB
Memory:	256 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

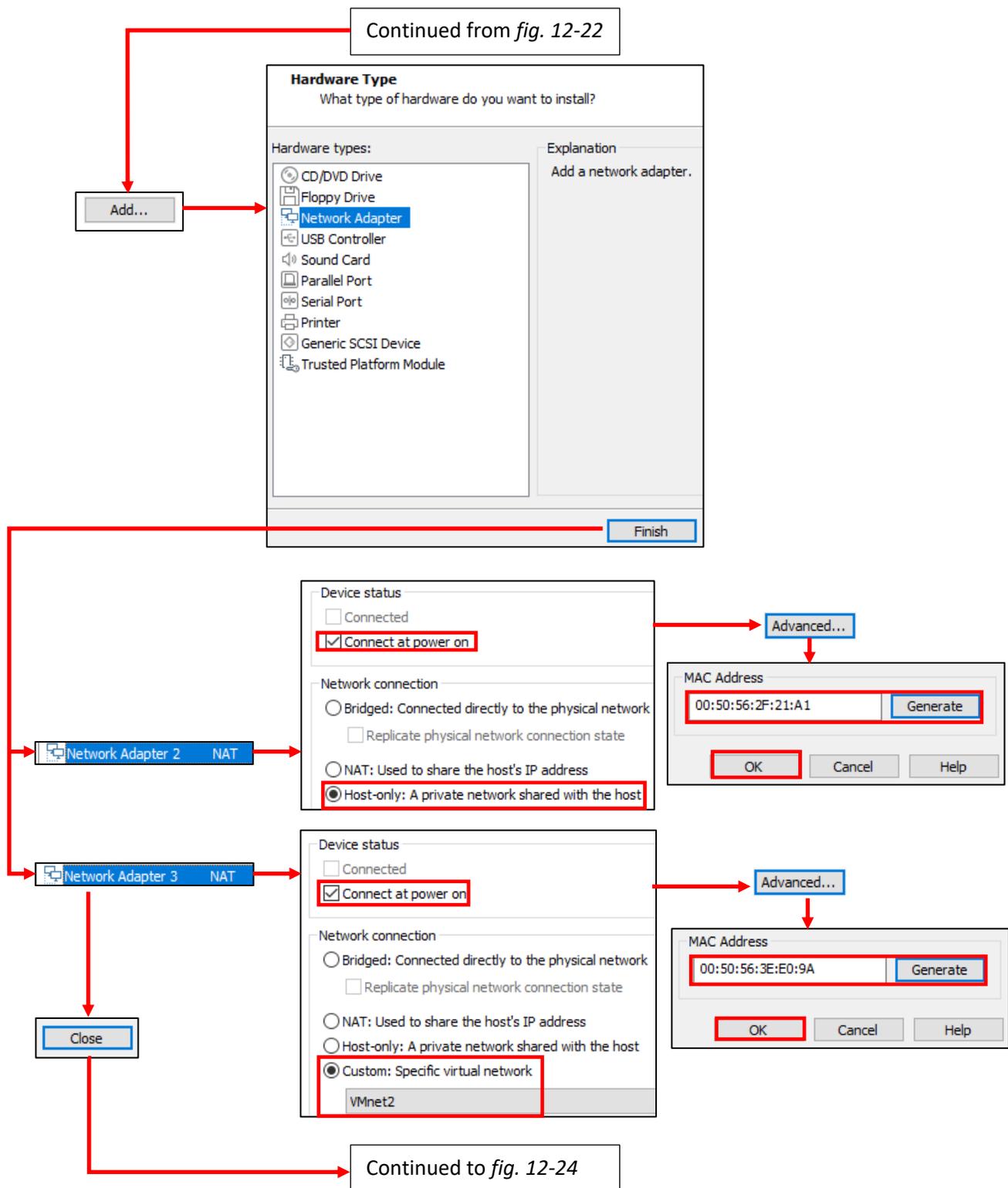
Customize Hardware...

Continued to *fig. 12-22*

12-21: On the *Specify Disk Capacity* screen, enter 5.0 into the *Maximum disk size (GB)* input box, and click the *Store virtual disk as a single file* radio button, then proceed to the *Ready to Create Virtual Machine* screen. Instead of clicking *Finish*, click the *Customize Hardware* button.



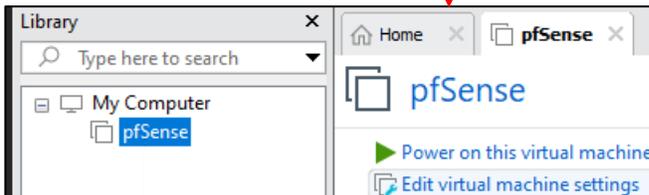
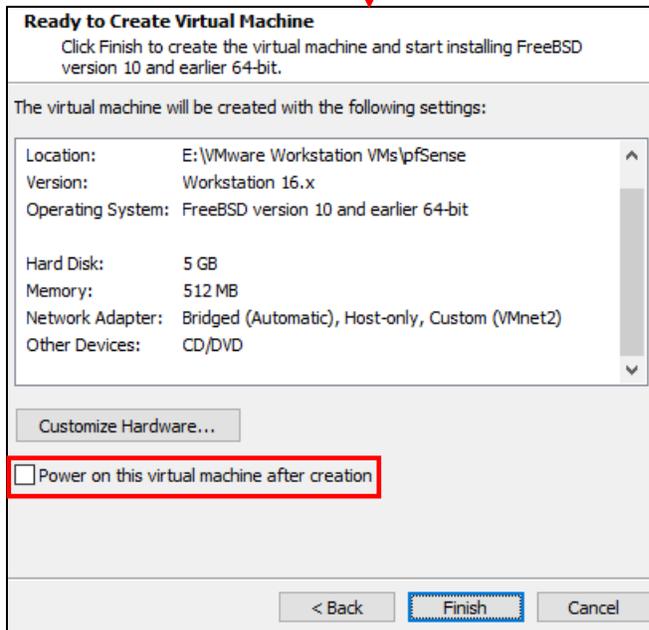
12-22: Welcome to the *Hardware* window. Begin by clicking on *Memory* and adjusting the *Memory for this virtual machine* input box to 512 (MB). Next up, click on the *Network Adapter* entry and adjust the *Network connection* radio button to *Bridged: Connected directly to the physical network*, and confirm the Device status checkbox *Connect at power on* is selected. Click on the *Advanced* button, and on the *Network Adapter Advanced Settings* window, ***Click the Generate button and record the MAC address that appears***, then Click the *OK* button to exit. Afterwards, highlight the *Sound Card* entry, then click *Remove*. Repeat this process for the *USB Controller* virtual hardware as well.



12-23: Next, click the *Add* button and use the *Add New Hardware Wizard* to add two new network cards – *Network Adapter 2* and *Network Adapter 3*. Adjust the *Network Connection* and *Advanced* settings for both adapters.

For *Network Adapter 2*, Select the *Host-only: A private network shared with the host* radio button. For *Network Adapter 3*, Select the *Custom: Specific virtual network* radio button, then select *VMnet2* (Linux users: `/dev/vmnet2`) from the drop-down list. For both network cards, confirm the *Connect at power on* checkbox is selected, then click the *Advanced* button, and generate a MAC address. **Record the MAC address generated for both newly added network interfaces.** Click the *Close* button to exit the *Hardware* window.

Continued from *fig. 12-23*



12-24: Finally, students come back to the *New Virtual Machine Wizard*. The pane in the center of the *Ready to Create Virtual Machine* window should reflect all of the changes made to the virtual machine. Students should confirm that their pfSense virtual machine's settings match what appears in the central pane. When finished, uncheck the *Power on this virtual machine after creation* checkbox, and click *Finish* to complete the wizard. After a moment or two, the pfSense VM should appear under the *Library* pane on the left side of the main window.

Noting the Notable

I can't overstate the value of documenting your lab network properly. Use whatever note-taking methods you prefer – paper and pen, Evernote, text editors, personal wikis, databases, spreadsheets, etc. Document the name of the VM, Operating system, the number of CPU cores allocated, RAM, Disk size, number of network adapters, network segments they are attached to, and their MAC addresses. This is called *asset management*, and it's an important habit to cultivate. Here is a template you can use for documenting your VMs:

VM Name:
Operating System:
CPU Cores:
RAM:
Disk Size:
Virtual Network Adapters:
Network Adapter #:
-Network Segment:
-MAC Address:
<Repeat for each network adapter>
Additional Notes:

And as an example, here is my pfSense VM entry:

VM Name: pfSense
Operating System: pfSense (FreeBSD)
CPU Cores: 1
RAM: 512MB
Disk Size: 5GB
Virtual Network Adapters: 3
Network Adapter 1:
-Network Segment: Bridged
-MAC Address: 00:50:56:3A:8D:C4
Network Adapter 2:
-Network Segment: Host-only (LAN)
-MAC Address: 00:50:56:2F:21:A1
Network Adapter 3:
-Network Segment: VMnet2 (IPS1)
MAC Address: 00:50:56:3E:E0:9A
Additional Notes: Lab firewall. Provides NTP, DNS, DHCP,
and HTTP proxy services.

Do this for every single virtual machine you add to your lab environment. Keep track of systems added or removed from the lab network. Always be aware of what's running on your networks. If you can do these things, you'll be better at asset management than most of the Fortune 500.

12.5.2 First Boot and OS Installation

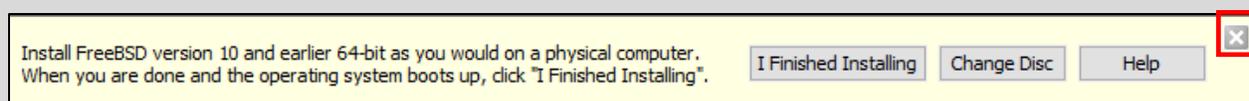
Our virtual machine has been created, and it is now time to install the pfSense operating system to the new VM. To begin, students will need to power on the virtual machine, then connect to its virtual console. The easiest way to do this is to click on the pfSense entry in the *Library* pane to highlight it, then click on *Power on this virtual machine*. Alternatively, students can right click on the pfSense VM in the *Library* pane, then select *Power > Power On*. The tab labeled pfSense in the VMware main window will change to display a virtual console session to the pfSense VM.

Think of this window as a direct keyboard, video, and mouse connection to the virtual machine while it is running. You'll notice a lot of text flying by as the VM boots from the installation ISO. Eventually you will reach the pfSense Installer. The first screen shows the *Copyright and distribution notice* for the software. Click anywhere in the virtual console window, and hit Enter to accept the software terms and conditions

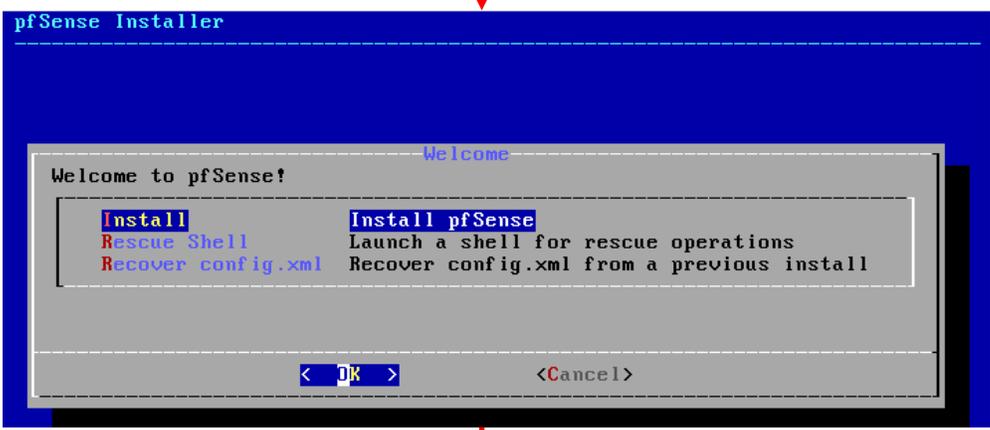
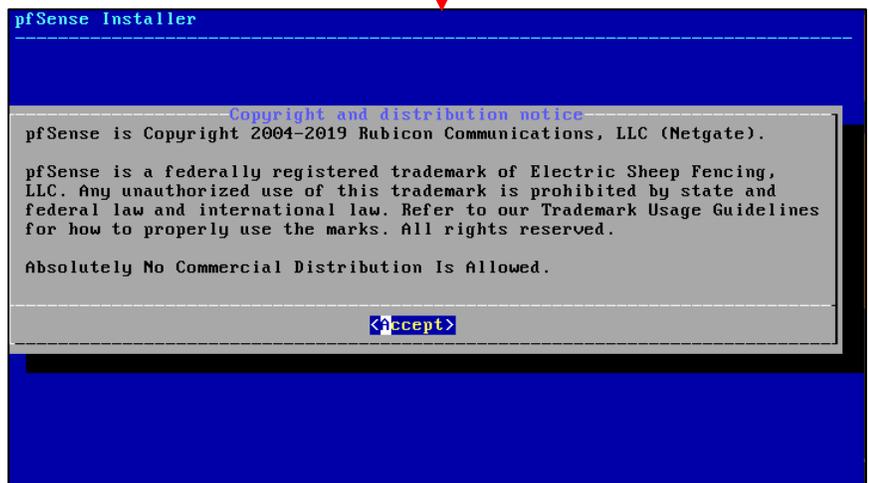
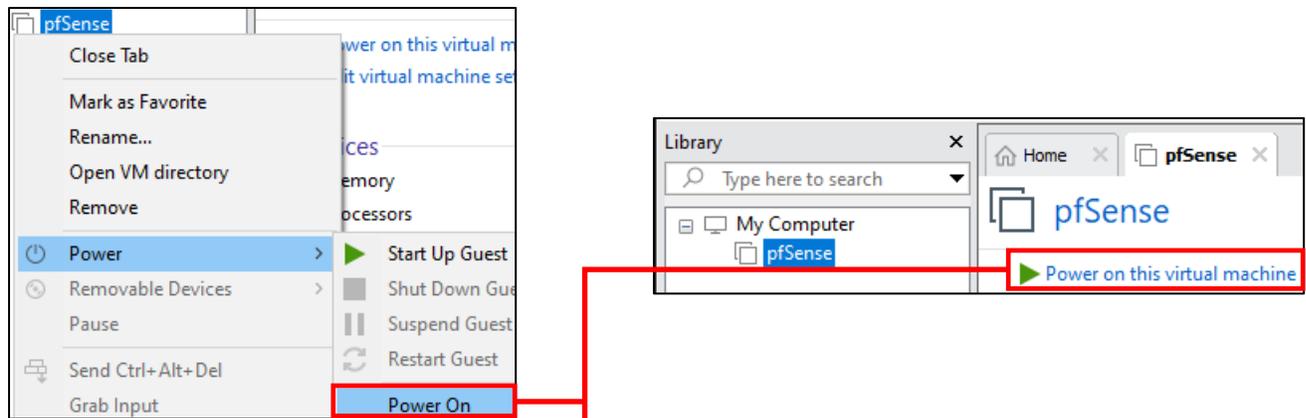
Next is the *Welcome* screen for the OS Installer. The option *Install pfSense* should be highlighted by default, but if not, use the arrow keys on your keyboard to select it, then hit enter. The next screen, titled *Keymap Selection* appears. If students are from a region of the world with a unique keyboard layout, they will need to search for and select it. Otherwise, select *Continue with default keymap* to use the US keymap, and hit enter. Next is the *Partitioning* screen. Partitioning is used to tell the installer how much and what portion of the disk to allocate. Since this is a virtual machine, and the disk is relatively tiny (5GB), select *Auto (UFS) Guided Desk Setup* and press enter to tell the installer to use all of the available disk space.

The installer handles formatting the disk and copying the operating system files over. The next screen, titled *Manual Configuration* asks if students want a command shell to manually edit any operating system files before closing the installer. Select *No*, and hit enter again. Finally, on the *Complete* screen, select the *Reboot* option and hit enter. While the VM is rebooting, hit the ctrl+alt keyboard combination to release control of the keyboard and mouse back to the host operating system, right-click on the pfSense VM entry in the *Library* pane once more, then select *Power > Power Off*. VMware Workstation will ask students to confirm if they wish to power off the virtual machine. Click the *Power Off* button to proceed (and optionally the *Do not show this message again* checkbox).

Note: Students may see a small notification along the bottom of the virtual console that reads: *Install FreeBSD version 10 and earlier 64-bit as you would on a physical computer. When you are done and the operating system boots up, click "I Finished Installing"*. As always, the hypervisor is wrong. Disregard these instructions, and click the small grey square with an X in the middle to remove this pop-up. Its very likely you'll see this pop when setting up the other VMs as well.



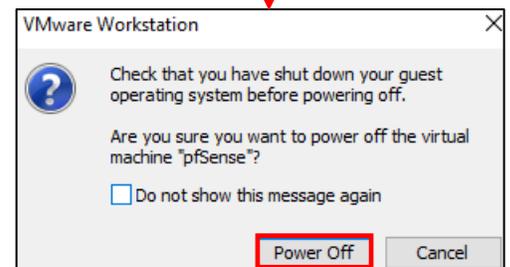
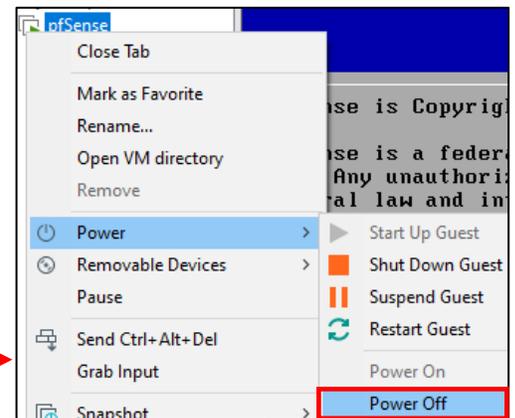
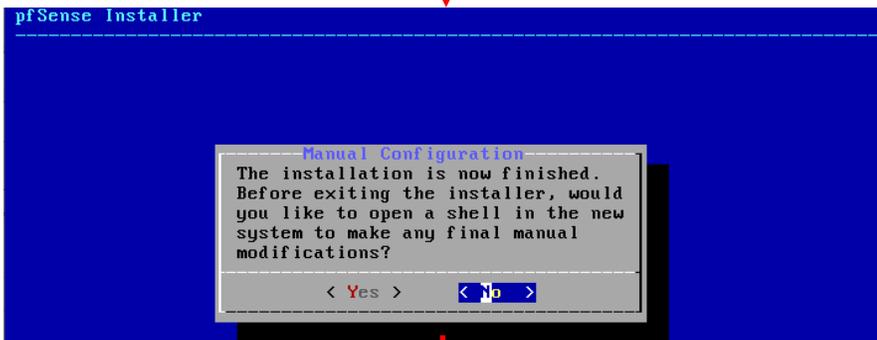
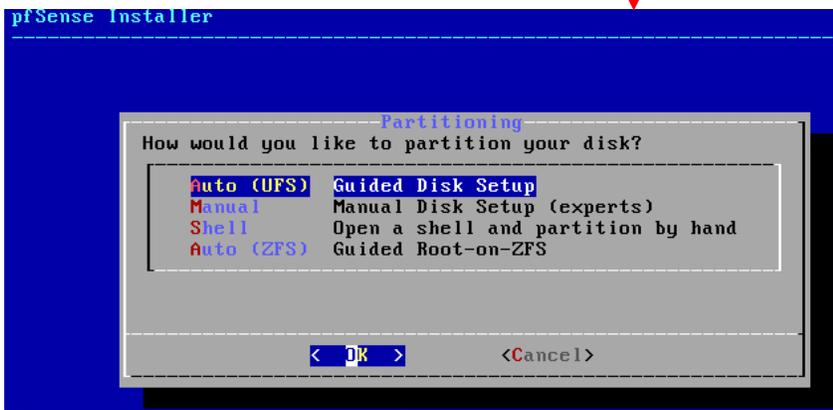
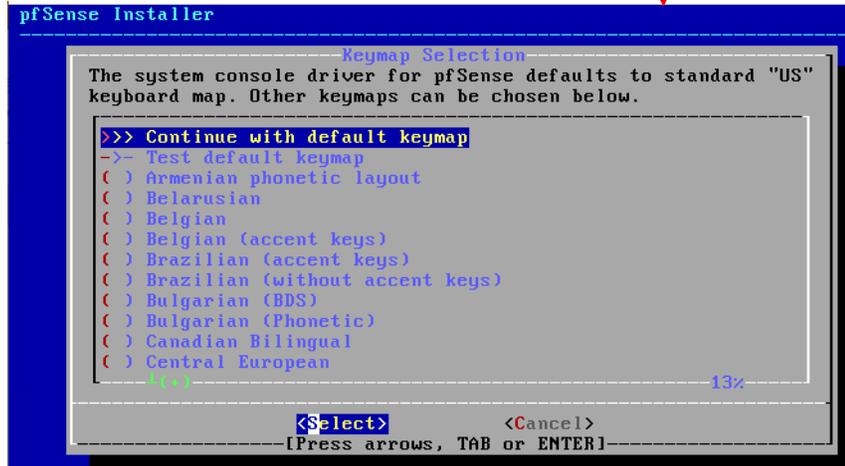
12-25: This dialogue box can safely be ignored. Click the gray square with the white X in the center to remove it.



Continued to fig. 12-27

12-26: Power on the newly created pfSense VM. VMware workstation should automatically connect students to the virtual console. The VM should boot from the installation ISO automatically. Accept the License Agreement, then select the *Install pfSense* option to proceed.

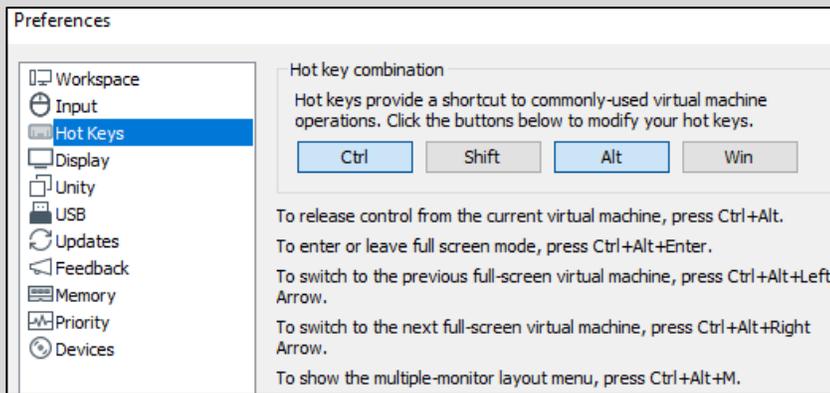
Continued from *fig. 12-26*



12-27: The rest of the installation process is pretty straightforward. Most students will be able to hit enter the entire way through and accept the defaults. Upon reaching the *Complete* screen, select the option to Reboot the virtual machine, and while the VM is rebooting, power it off.

Virtual Machines Ate My Neighbors Input

When a user clicks on the virtual machine console, that window grabs all of the input from the mouse and keyboard. That means that if there are other applications running on your host system you want to interact with, you have to tell the virtual console to let go of the mouse and keyboard first. VMware Workstation uses a special key binding called the *Hot key combination* to signal to the virtual console that you have other things to do. On Windows and Linux, this is the Ctrl+Alt key combination. This keyboard combination can be found and modified in *Edit > Preferences > Hot Keys*.

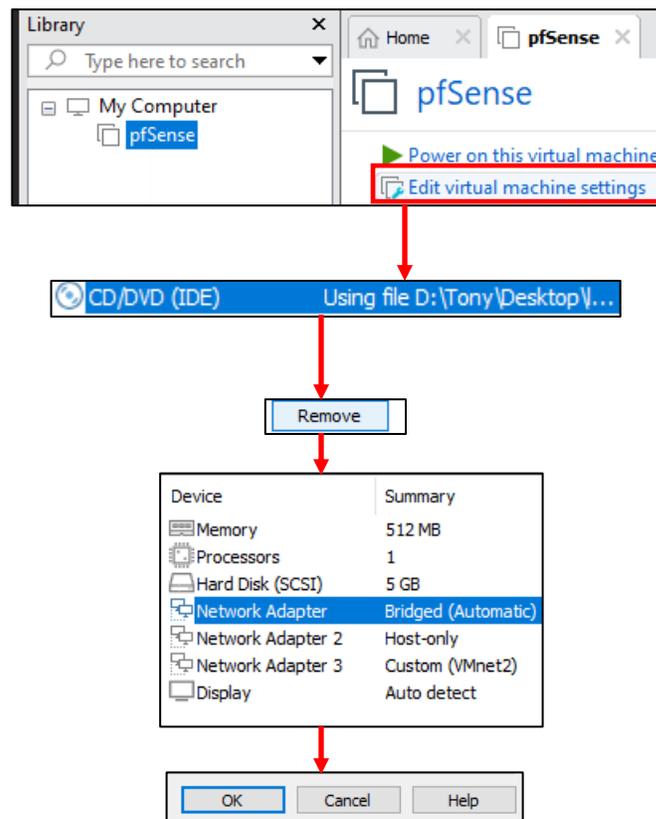


12-28: Look for the *Hot key Combination* option under *Edit > Preferences > Hot Keys* if you need to know what key releases control from the virtual console, or to rebind the shortcut combo.

12.5.3 Virtual Machine Settings

Before students can get started customizing the pfSense operating system for use in their lab environment, there is one more minor edit that needs to be made: Removing the Virtual CD/DVD drive. In the main VMware Workstation window, on the left pane labeled *Library*, left click on the pfSense VM entry to highlight it, then click on *Edit virtual machine settings* to open up a window labeled *Virtual Machine Settings*. This window will have two tabs labeled *Hardware* and *Options*. The *Hardware* tab should already be selected by default, but if it is not for some reason, select it. Students should be greeted with a window that looks almost identical to the Hardware window from the *New Virtual Machine Wizard*.

In the *Device* column on the left pane, click on the *CD/DVD (IDE)* entry to highlight it, then click the *Remove* button all the way at the bottom of the left pane. Once finished with this task, Click the *OK* button towards the bottom right corner of the window to close the *Virtual Machine Settings* window.



12-29: Click on pfSense in the *Library* pane, followed by *Edit virtual machine settings*. The *Virtual Machine Settings* window opens. In the *Hardware* tab, highlight *CD/DVD (IDE)*, then click *Remove*. When finished, review the list of devices installed on the virtual machine. Students' device listing should look nearly identical to what is displayed above. Click *OK* to exit.

12.5.4 pfSense Command-Line and initial interface configuration

In this section, readers will navigate the command-line interface of their pfSense virtual machine to perform essential setup tasks. Once completed, users can navigate to the webConfigurator interface. Begin by powering on the pfSense VM. After a few moments, the boot process completes and students are presented with the pfSense command-line menu. This menu features a series of configuration and troubleshooting options. Begin by selecting option 1 and hitting enter.

12.5.4.1 The Assign Interfaces Wizard

Users are greeted by the *Assign Interfaces* wizard. This wizard is used to map our virtual machine's network interfaces (*Adapter 1*, *Adapter 2*, and *Adapter 3*) to their pfSense aliases – *WAN*, *LAN*, or *OPT1*. Unfortunately, the operating system itself also has unique names for each of these interfaces, adding another layer of complexity and confusion when trying to perform this task.

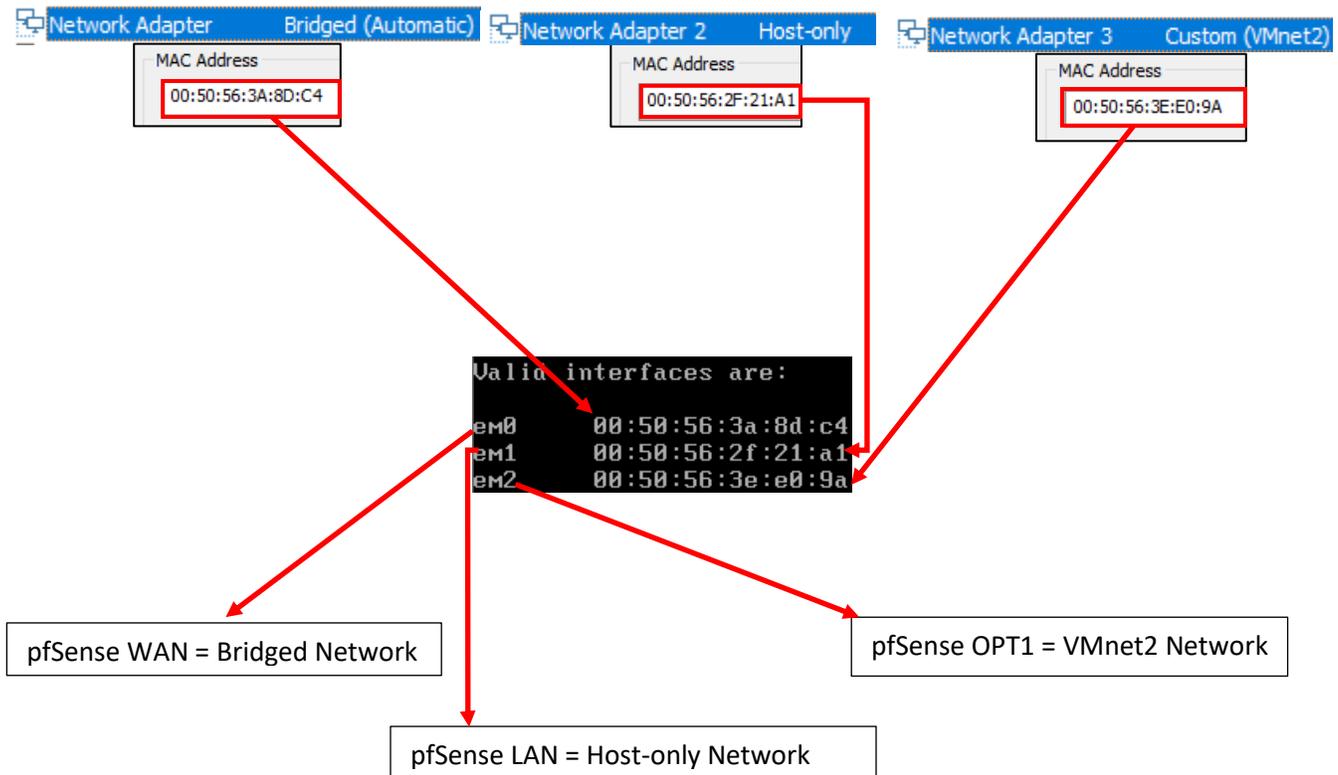
pfSense itself is based on the FreeBSD operating system, and BSD has its own methods for assigning physical (or virtual, in our case) network interfaces an interface name. For example, BSD assigned the network adapters of my virtual machine the interface names *em0*, *em1*, and *em2*. Every network adapter – integrated or not, virtual or physical, wired or wireless – all have a MAC address to uniquely identify them on a local network. We're going to take advantage of that to know for certain which of the three interfaces *em0* through *em2* map to *Network Adapters 1* through *3*, and how they should be assigned as the *WAN*, *LAN* and *OPT1* aliases. Students were highly advised to record the contents of the *Mac Address* input box of *Network Adapters 1*, *2* and *3* to assist in this task.

A quick way for readers to determine the interface names for their pfSense installation is through the wizard itself. Upon selecting option 1, a section of text labeled *Valid interfaces are* appears, followed by a series of lines. ***Students should have 3 of these lines in total.*** These lines provide the interface names, MAC addresses, current operational status, and type of hardware BSD identifies the network interface as (The drivers BSD loaded) for each network interface pfSense was able to detect. Here is an example:

```
Valid interfaces are: 2
1
em0      00:50:56:3a:8d:c4
em1      00:50:56:2f:21:a1
em2      00:50:56:3e:e0:9a
```

12-30: A portion of the *Assign Interfaces* wizard. Pay attention to the interface names (1) and the MAC addresses for those interface names (2). This information is needed to determine which virtual network segment they are connected to. This in turn allows students to assign the *WAN*, *LAN* and *OPT1* interfaces correctly.

Compare the MAC addresses displayed, to the MAC addresses recorded earlier, and use that information to complete the rest of the Assign Interfaces wizard. A diagram (fig. 12-31) is provided below to help students understand how to correctly perform this mapping process.



12-31: Here we have the network configuration for my pfSense VM, and the output from the valid interfaces table from the *Assign Interfaces* wizard. *Adapter 1* has the MAC Address 00:50:56:3A:8D:C4. Looking at the valid interfaces table, *em0* has the same MAC address. This means that *em0* maps to *Network Adapter 1*, connected to the bridged network. *em0* should be assigned as the *WAN* interface. *Adapter 2*'s MAC address matches the MAC address for *em1*. This means *em1* maps to *Network Adapter 2*, connected to the host-only network – our management network. This means *em1* should be assigned the *LAN* interface. Finally, *Network Adapter 3* matches the MAC address for *em2*. This means *em2* maps to the *VMnet2* – IPS 1. This means that *em2* should be assigned the *OPT1* interface.

The remainder of this section will aim to guide students through the various questions the wizard will ask (in *italicized* font), and the answers I provided (in *bold* font) based on my lab network and adapter to MAC address mappings. **Students should be aware that this is by and far the most important configuration task for pfSense.** Making sure that the VirtualBox adapters map to the correct pfSense aliases and network segments is absolutely vital to the lab environment working correctly.

Should VLANs be set up now [y|n]? n

*Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): **em0***

*Enter the LAN interface name or 'a' for autodetection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): **em1***

*Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): **em2***

The interfaces will be assigned as follows:

WAN -> em0

LAN -> em1

OPT1 -> em2

*Do you want to proceed [y|n]? **y***

After answering these questions, pfSense will loop back to the main menu.

```

pfSense 2.4.5-RELEASE amd64 Tue Mar 24 15:25:50 EDT 2020
Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)
VirtualBox Virtual Machine - Netgate Device ID: 5a25c089ce30ec1b5b9e

*** Welcome to pfSense 2.4.5-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.0.39/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:
/64
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults    13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option: 1

Valid interfaces are:

em0      00:50:56:3a:8d:c4   (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em1      00:50:56:2f:21:a1   (up) Intel(R) PRO/1000 Legacy Network Connection 1.
em2      00:50:56:3e:e0:9a (down) Intel(R) PRO/1000 Legacy Network Connection 1.

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

The interfaces will be assigned as follows:

WAN -> em0
LAN -> em1
OPT1 -> em2

Do you want to proceed [y|n]? y

```

12-32: A selection of screen captures from the Assign Interfaces wizard, stitched together to show the questions the wizard asks, and the responses based on network adapter mappings in *fig. 12-31*.

12.5.4.2 Setting IP Addresses for WAN, LAN, and OPT1

The next task we will need to perform on the pfSense command-line is assigning IP addresses to the WAN, LAN, and OPT1 interfaces using the *Set interface(s) IP address* wizard. Most students will have their host system connected to a home or enterprise network where DHCP is available, and just about anything that requests an IP address lease will get one with no problems. That *should* include the pfSense WAN interface bridged to that network. This means the WAN interface should already have an IP address, subnet mask, default gateway (and usually, DNS servers to forward DNS requests to) automatically provided (if this is not the case, see the sidebar discussion, *Help! The WAN Interface has no IP Address*, for some troubleshooting pointers). That means we should only have to run through the Set interface(s) IP address wizard twice – once for the LAN interface, and once for the OPT1 interface. Select option 2 from the pfSense menu to get started.

Similar to the previous section (12.5.4.1), the remainder of this section is going to consist of the questions the *Set interface(s) IP address* wizard will ask students (*italicized*), and the correct answers for the LAN and OPT1 interfaces (in **bold**), followed by an illustration depicting the same questions and answers.

LAN interface:

Available interfaces:

- 1 – WAN (*[interface name] – [dhcp/dhcp6/static address configuration]*)
- 2 – LAN (*[interface name] – static*)
- 3 – OPT1 (*[interface name]*)

Enter the number of the interface you wish to configure: **2**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.1.1**

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense

e.g. 255.255.255.0 = 24

255.255.0.0 = 16

255.0.0.0 = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.1.10**
Enter the end address of the IPv4 client address range: **172.16.1.254**
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24

You can now access the webConfigurator by opening the following URL in your web browser:

<https://172.16.1.1>

Press <ENTER> to continue. <ENTER>

```
Enter the number of the interface you wish to configure: 2
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 172.16.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
    255.255.0.0   = 16
    255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.1.10
Enter the end address of the IPv4 client address range: 172.16.1.254
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
    https://172.16.1.1/

Press <ENTER> to continue.
```

12-33: Screen captures from the *Set interface(s) IP address wizard*, stitched together to show the questions the wizard asks, and the responses for the LAN interface based on network adapter mappings in *fig. 12-31*.

Here is an abridged set of *questions* and **answers** for the *OPT1* interface:

OPT1 interface (abridged):

Available interfaces:

- 1 – WAN ([*interface name*] – [*dhcp/dhcp6/static address configuration*])
- 2 – LAN ([*interface name*] – *static*)
- 3 – OPT1 ([*interface name*])

Enter the number of the interface you wish to configure: **3**

Enter the new LAN IPv4 address: Press <ENTER> for none:
> **172.16.2.1**

Enter the new LAN IPv4 subnet bit count (1 to 31):
> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:
> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.2.10**
Enter the end address of the IPv4 client address range: **172.16.2.254**

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

```

Enter the number of the interface you wish to configure: 3

Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 172.16.2.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new OPT1 IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new OPT1 IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on OPT1? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.2.10
Enter the end address of the IPv4 client address range: 172.16.2.254

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to OPT1...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 OPT1 address has been set to 172.16.2.1/24

Press <ENTER> to continue.

```

12-34: Screen captures from the *Set interface(s) IP address* wizard, stitched together to show the questions the wizard asks, and the responses for the *OPT1* interface based on network adapter mappings in *fig. 12-31*.

After running the wizard again for the *OPT1* interface, students should have an IP address for the *WAN*, *LAN* and *OPT1* interfaces. Additionally, DHCP ranges should be assigned for the *LAN* and *OPT1* interfaces. We're just about ready to move to the webConfigurator, but before doing so, lets run some network connectivity tests first.

```

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.0.197/24
                v6/DHCP6: 2601:408:502:c330:a00:27ff:fe23:b366
/64
LAN (lan)     -> em1      -> v4: 172.16.1.1/24
OPT1 (opt1)   -> em2      -> v4: 172.16.2.1/24

```

12-35: The interface information portion of the pfSense command-line menu should look something like this. Looking good is one thing, now let's see if it actually works.

What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?

Unfortunately, I have no way of knowing what network ranges students use at home, so it's entirely possible your physical network may already be using one of the ranges I'm asking you to configure for your lab environment (e.g. 172.16.1.0/24, or 172.16.2.0/24). **To avoid network conflicts on your home network, maybe try these alternate configurations for the *Set interface(s) IP address wizard*:**

Alternate LAN configuration:

LAN interface IP address: 172.16.11.1
Subnet mask bit count: 24
DHCP start address: 172.16.11.10
DHCP end address: 172.16.11.254

Alternate OPT1 configuration:

OPT1 interface IP address: 172.16.12.1
Subnet mask bit count: 24
DHCP start address: 172.16.12.10
DHCP end address: 172.16.12.254

If your lab network is connected to a school or enterprise network using the entire 172.16.0.0/12 allocation, things may be a little more complicated. It may be best to use one of the other RFC1918 network allocations instead, such as 192.168.0.0/16, or 10.0.0.0/8. Why? Enterprise networking can become complicated, either due to growth over time, legacy configurations, or work-arounds to problems accrued over time. You don't want to troubleshoot network problems on your host system, nor do you want the IT ops team coming to your desk over a network outage that could've been avoided. **Here are some alternate configurations for the *Set interface(s) IP address wizard* if you need to avoid using 172.16.0.0/12 entirely:**

Alternate LAN configuration 1:

LAN interface IP address: 10.0.11.1
Subnet mask bit count: 24
DHCP start address: 10.0.11.10
DHCP end address: 10.0.11.254

Alternate OPT1 configuration 1:

LAN interface IP address: 10.0.12.1
Subnet mask bit count: 24
DHCP start address: 10.0.12.10
DHCP end address: 10.0.12.254

Alternate LAN configuration 2:

LAN interface IP address: 192.168.11.1
Subnet mask bit count: 24
DHCP start address: 192.168.11.10
DHCP end address: 192.168.11.254

Alternate OPT1 configuration 2:

LAN interface IP address: 192.168.12.1
Subnet mask bit count: 24
DHCP start address: 192.168.12.10
DHCP end address: 192.168.12.254

Substituting Instructions for Your Chosen Network Ranges

Keep in mind you don't have to use the alternate configurations recommended above. If students have some experience with networking and subnetting, they're welcome to use any network range that suits them. These are just some suggestions to help those who are not quite as experienced, and want to avoid network conflicts.

As a final reminder, **the remaining sections, chapters, and configuration steps will all assume that readers are using 172.16.1.0/24 for the LAN network and 172.16.2.0/24 for the OPT1 network.** This means you will have to mentally substitute steps and commands for the network range you are using instead.

For example, the lab network diagram in chapter 6 has the Kali VM on the IPS 1 (OPT1) network, with an IP address of 172.16.2.2. If you are using an alternate network configuration for the OPT1 network, say 192.168.12.0/24, then the Kali VM's IP address should be 192.168.12.2. If I say "*run the command ssh username@172.16.2.2 to connect to the kali VM*", you'll have to mentally substitute that with `ssh username@192.168.12.2` instead. As another example, firewall rules denying access to or from 172.16.2.3 (Metasploitable2) should be created for 192.168.12.3 instead. Keep this in mind as you continue to build your lab network!

Help! The WAN Interface has no IP Address

If the WAN interface of your pfSense VM has no IP address, consider some of the following to help with troubleshooting:

-No DHCP – It's pretty rare, but perhaps the WAN interface is bridged to a network without DHCP. This just means that you'll have to run the *Set interface(s) IP address* wizard to manually configure the WAN interface IP address, subnet mask, and default gateway. I've already listed the questions the wizard asks, and provided the answers for the LAN and OPT 1 interfaces, but since I have absolutely no idea what IP address and subnet mask is assigned to your local physical network, I cannot tell you what you need to enter for the wizard.

If you don't know either, ask a network administrator or whoever is responsible for your network to assist you. Note that if required to manually configure these settings here, practically all of the tasks that require DNS to be configured (e.g., network connectivity tests, and checking for updates on your VMs) will not work until DNS server addresses are configured. This can be done via the webConfigurator, and will be covered shortly.

-Bridged to the wrong host adapter – Another possibility is perhaps the WAN interface virtual adapter may be bridged to the wrong physical network adapter on your host system. If you suspect this is the case, check out [section 12.3](#) (pp. 416-422) for information on how to access the *Virtual Network Editor*, and edit which network interface(s) the Bridged network segment connects to. Then check out [section 12.5.4.1](#) (pp. 442-445) to make absolutely sure that the pfSense WAN interface was mapped to the correct VirtualBox network interface.

-NAC Interference – If you are network security enthusiast at home or connected to an enterprise network, NAC (network access control) may be preventing the WAN interface from obtaining an IP address. Back in chapter 4, [section 4.1.2, NAT Networking \(and Port Forwarding\)](#) (pp. 46-47), readers learned how network address translation works, and how in situations like this, you may be forced to use VMware Workstation's NAT network options to work around network security. If you suspect the WAN interface is being blocked, you can try editing the pfSense VM's *Network* settings for the adapter attached to the bridged network, and instead attach it to the NAT network. For example, [fig. 12-30](#) shows *Network Adapter* is attached to the *Bridged* network. Open the *Virtual Machine Settings* menu, Highlight *Network Adapter*, select *NAT*, then click *OK*. Reboot the pfSense VM. It should have an IP address from the *NAT* network, but that doesn't mean it has network connectivity. Continue below to the next section as normal to test your network connectivity.

Note: If this doesn't work, or attempting to subvert network access controls would otherwise get students in trouble, consider talking to your network/systems administrator and seeing if you can get DHCP allocation and/or necessary exceptions put into place. **Don't violate acceptable use policies, and don't break the law.**

12.5.5 Testing Internet Connectivity using Shell commands

Select option 8, labeled *Shell* in the pfSense menu. Doing so will open up a command-line (bash) shell. Run these 3 commands, and observe their output:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

Here is output from these 3 commands:

```
Enter an option: 8

[2.4.5-RELEASE][root@pfSense.localdomain]/root: ping -c 4 www.google.com
PING www.google.com (172.217.6.100): 56 data bytes
64 bytes from 172.217.6.100: icmp_seq=0 ttl=54 time=24.496 ms
64 bytes from 172.217.6.100: icmp_seq=1 ttl=54 time=22.714 ms
64 bytes from 172.217.6.100: icmp_seq=2 ttl=54 time=21.638 ms
64 bytes from 172.217.6.100: icmp_seq=3 ttl=54 time=19.490 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 19.490/22.084/24.496/1.813 ms
[2.4.5-RELEASE][root@pfSense.localdomain]/root: nslookup www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.6.100
Name:   www.google.com
Address: 2607:f8b0:4009:812::2004

[2.4.5-RELEASE][root@pfSense.localdomain]/root: curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Mon, 01 Jun 2020 02:53:41 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Mon, 01 Jun 2020 02:53:41 GMT
cache-control: private
set-cookie: 1P_JAR=2020-06-01-02; expires=Wed, 01-Jul-2020 02:53:41 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=DdNU16afHrYu25Utm83temwvvrSe6a4UyA3YHz_JKLFzBAv7xrWi8HjSn2-x1PNmxh3EutjAoFBh15hNpxrU72.jpzLLQU0JHJxaOMh5mFyntk5Gae7KUMeZ-d1g8I1KloIb7HzOBP_BB4b0sb4lt0Tv1zwOdriVE8ndqfygcrN04; expires=Tue, 01-Dec-2020 02:53:41 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

12-36: The output from the `ping -c 4`, `nslookup`, and `curl -I` commands. All three of these commands completed successfully. Pay close attention to the marked sections above. Note that the IP addresses returned for `nslookup` (In the fields labeled *Address*) may vary based on region.

In a nutshell, these three commands are being used to test various forms of internet connectivity for our VM. `ping -c 4 www.google.com` tells pfSense to send 4 (and only 4) ICMP packets to a specific destination, requesting that the destination respond with its own ICMP packets if it has been reached. `nslookup www.google.com` asks our pfSense virtual machine's configured DNS servers to translate a domain name to an IP address for us. Finally, `curl -I https://www.google.com` is being used to test HTTPS connectivity to the internet. The `-I` option tells the command to only return the HTTP Server headers from our request. All we're really interested in is the line of text: `HTTP/2 200`. This is a thumbs up from Google's webserver confirming that they got our HTTP request with no problems.

Students already familiar with DNS basics may have noticed that we are already trying to ping a domain name (`www.google.com`) with our `ping` command. This means, that in order to actually ping the correct destination, our virtual machine will need to make a DNS request to find the IP address of `www.google.com`. That makes the `nslookup` test redundant, right? Well, yes and no. Later in this chapter, as new virtual machines get created, readers will be advised to perform connectivity tests on those VMs as well. However, the pfSense firewall policy is going to be very strict, so ICMP packets outbound from our lab network will be blocked. Due to how DNS works, the `nslookup` check can still be used to make sure VMs can resolve domain names, and the `curl` connectivity test will be more than sufficient to confirm whether or not lab virtual machines have the internet access they require. After performing these commands and confirming internet connectivity, type `exit` to leave the shell.

My connectivity commands failed! Now what?

If students got anything other than output similar to *fig. 12-36* (e.g., request timeouts and/or packet loss for `ping`, timeouts for `nslookup`, no response for `curl -I`), then there are connectivity issues to be sure. Troubleshooting network connectivity is an extremely complex topic. I can't give you a definitive guide for finding the root of your problem, but I can tell you to start with the basics and work your way up – sometimes the cause of your network problems are settings or hardware that was taken for granted.

Checking physical cabling, link lights and physical connectivity to network devices always comes first. As an extension to that, check out the sidebar in section 12.5.4.2 (*Help! The WAN interface has no IP address*) for some additional clues. The VM may be bridged to the wrong physical adapter. Some form of network security (e.g., a network firewall) may be preventing your VM from connecting to the internet. Try connecting the bridged adapter to a NAT network instead. The incorrect network adapter may have been chosen to be the WAN interface. Consider re-running the *Assign Interfaces* wizard again, and compare the MAC addresses from the wizard to the MAC addresses of the network adapter in the pfSense VM's Settings menu. See *fig. 12-31* for guidance on confirming that interfaces have been mapped correctly.

If students were required to run the *Set interface(s) IP address* wizard for the WAN interface (No DHCP), or your local network's DHCP server doesn't assign DNS servers automatically, your troubleshooting commands will fail because pfSense has no way of resolving domain names. We will be covering how to manually configure a primary and/or secondary DNS server for pfSense via the webConfigurator shortly.

If your host system is connected to a physical network already using 172.16.1.0/24 or 172.16.2.0/24, you may be experiencing network conflicts, routing loops, or other weird behavior. Assign different IP addresses and ranges to the LAN and OPT1 networks to avoid network conflicts. See the sidebar discussion in 12.5.4.2 labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?*

Last but not least, check and double check that you entered the commands correctly. Typos matter on the command-line, and BSD will not hold your hand if the command is entered incorrectly. If all else fails, don't be afraid to ask others for guidance.

12.5.6 Finish setting up pfSense

Navigate to [chapter 14, *pfSense Firewall Policy and Network Services*](#), starting on p. 664 and follow the chapter guidance. Once completed, readers will be directed back here to complete their lab environment.

12.6 Create the Remaining Virtual Machines

Welcome back! Now that the pfSense VM is fully functional, it's time to start working on the remaining lab VMs. In this section, users will create three of the four remaining virtual machines via the *Create Virtual Machine* wizard, then adjust the *Settings* of each virtual machine. After the SIEM, IPS and Kali VMs are created and configured, readers will be guided through the operating system installation, and initial setup process for all three VMs. The Metasploitable 2 VM is a unique case, and will be covered separately.

12.6.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali

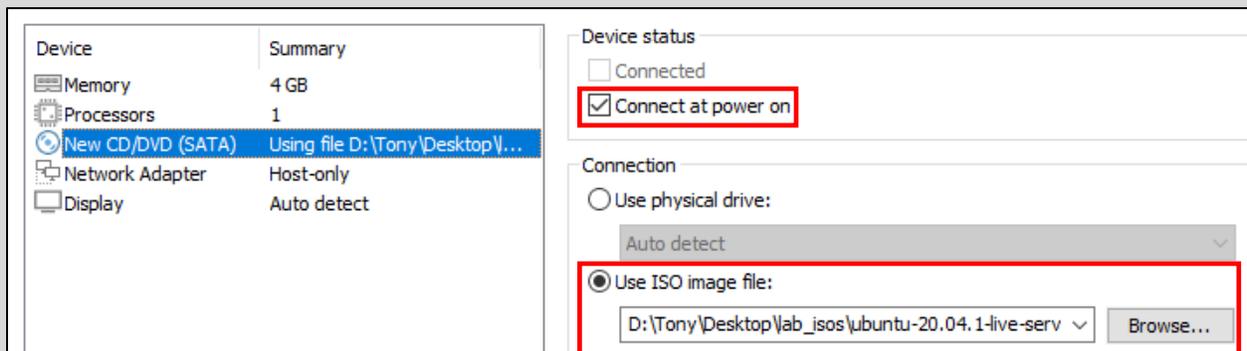
Run the *New Virtual Machine Wizard* three times, with the *Typical* radio button selected, and the settings listed in the table below. Assume the default for any settings not mentioned in the table below. Refer back to [section 12.5.1 \(pp. 427-436\)](#) for guidance on how to access and progress through the wizard as needed.

Guest Operating System Installation:	I will install the operating system later.	I will install the operating system later.	I will install the operating system later.
Guest Operating System:	Linux	Linux	Linux
Version:	Ubuntu 64-bit	Ubuntu 64-bit	Debian 10.x 64-bit
Virtual Machine Name:	SIEM	IPS	Kali
Specify Disk Capacity:	80.0 GB <i>Store virtual disk as a single file</i>	80.0 GB <i>Store virtual disk as a single file</i>	80.0 GB <i>Store virtual disk as a single file</i>
Customize Hardware:			
-Memory:	4GB (4096MB)	4GB (4096MB)	4GB (4096MB)
-Processors:	1 Processor 1 Core Total processor cores: 1	1 Processor 1 Core Total processor cores: 1	1 Processor 1 Core Total processor cores: 1
-New CD/DVD Drive (SATA):	Under <i>Connection</i> , select <i>Use ISO Image file</i> . Click <i>Browse</i> , locate the Ubuntu Server ISO, and select it. Ensure <i>Connect at power on</i> (under Device Status) is selected.	Under <i>Connection</i> , select <i>Use ISO Image file</i> . Click <i>Browse</i> , locate the Ubuntu Server ISO, and select it. Ensure <i>Connect at power on</i> (under Device Status) is selected.	Under <i>Connection</i> , select <i>Use ISO Image file</i> . Click <i>Browse</i> , locate the Kali Linux ISO, and select it. Ensure <i>Connect at power on</i> (under Device Status) is selected.
-Number of Network Adapters:	1	3 (Click <i>Add</i> button, select <i>Network Adapter</i> twice)	1
-Network Adapter Network Connections:	Network Adapter: Host-only	Network Adapter: Host-only Network Adapter 2: VMnet2 (/dev/vmnet2) Network Adapter 3: VMnet3 (/dev/vmnet3)	Network Adapter: VMnet2 (/dev/vmnet2)
-Remove the following virtual hardware:	USB Controller Sound Card Printer	USB Controller Sound Card Printer	USB Controller Sound Card Printer

Why am I choosing *I will install the operating system later*?

Students may be wondering why when they run the New Virtual Machine Wizard I'm telling you to select the *I will install the operating system later* option on the *Guest Operating System Installation* screen. It's because VMware Workstation has a "helpful" feature called *Easy Install*, that will make a lot of presumptions and do some things that we don't necessarily want the hypervisor to do for us. Namely, it installs VMware tools, and we absolutely don't want that to happen. The easiest way around this is to tell VMware that we'll install the operating system later.

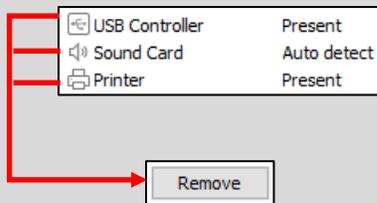
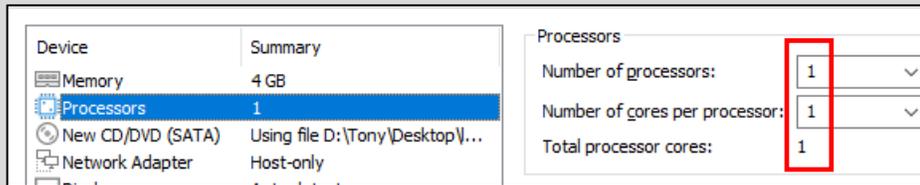
Then, using the *Customize Hardware* button in the *New Virtual Machine Wizard* (Or, if you accidentally clicked *Finish*, the Virtual Machine Settings window – which we went over how to access in section 12.5.1 in the sidebar discussion, *Jumped The Gun* on p. 429), students can highlight the *New CD/DVD Drive (SATA) Device*, select the *Use ISO image file* radio button, and browse to the location of the Ubuntu Server and/or Kali Linux ISO files. Also, ensure that the checkbox *Connect at power on* is checked.



12-37: To avoid having *Easy Install* make a mess of our virtual machines, select the *I will install the operating system later* option during the *New Virtual Machine Wizard*. Then, while you are customizing your virtual machines, modify the *New CD/DVD (SATA) device*. Select the *Use ISO image file* radio button and browse to the Ubuntu Server 20.04 ISO file (SIEM, IPS VMs) or the Kali Linux ISO (Kali VM) as appropriate. Ensure the *Connect at power on* checkbox is checked.

Other Assumptions (That are also Wrong)

VMware Workstation thinks its smarter than us, so when you tell the *New Virtual Machine Wizard* you want to install a Linux virtual machine, it makes some assumptions. Workstation assumes we want a VM with two CPU cores. This means that during the *Customize Hardware* phase, you'll want to select the *Processors Device* listing and set both the *Number of processors*, and *Number of cores per processor* to 1, so that *Total processor cores* reads 1 For all three virtual machines. Finally, in addition to removing both the *Sound Card* and *USB Controller* devices, students will need to remove a *Device* labeled Printer during the *Customize Hardware* portion. These are all reflected in the table above.



12-38: VMware workstation makes a couple of assumptions that we really don't want it to make when it creates Linux virtual machines. While you are customizing the hardware of all three virtual machines, Be sure to re-adjust the *Processors* device listing until the *Number of processors*, *Number of cores per processor*, and *Total processor cores* all read 1. Additionally, be sure to *Remove* the *USB Controller*, *Sound Card*, and *Printer* virtual hardware devices.

The virtual machine will be created with the following settings:	
Name:	SIEM
Location:	E:\VMware Workstation VMs\SIEM
Version:	Workstation 16.x
Operating System:	Ubuntu 64-bit
Hard Disk:	80 GB
Memory:	4096 MB
Network Adapter:	Host-only
Other Devices:	CD/DVD

The virtual machine will be created with the following settings:	
Name:	IPS
Location:	E:\VMware Workstation VMs\IPS
Version:	Workstation 16.x
Operating System:	Ubuntu 64-bit
Hard Disk:	80 GB
Memory:	4096 MB
Network Adapter:	Host-only, Custom (VMnet2), Custom (VMnet3)
Other Devices:	CD/DVD

The virtual machine will be created with the following settings:	
Name:	Kali
Location:	E:\VMware Workstation VMs\Kali
Version:	Workstation 16.x
Operating System:	Debian 10.x 64-bit
Hard Disk:	80 GB
Memory:	4096 MB
Network Adapter:	Custom (VMnet2)
Other Devices:	CD/DVD

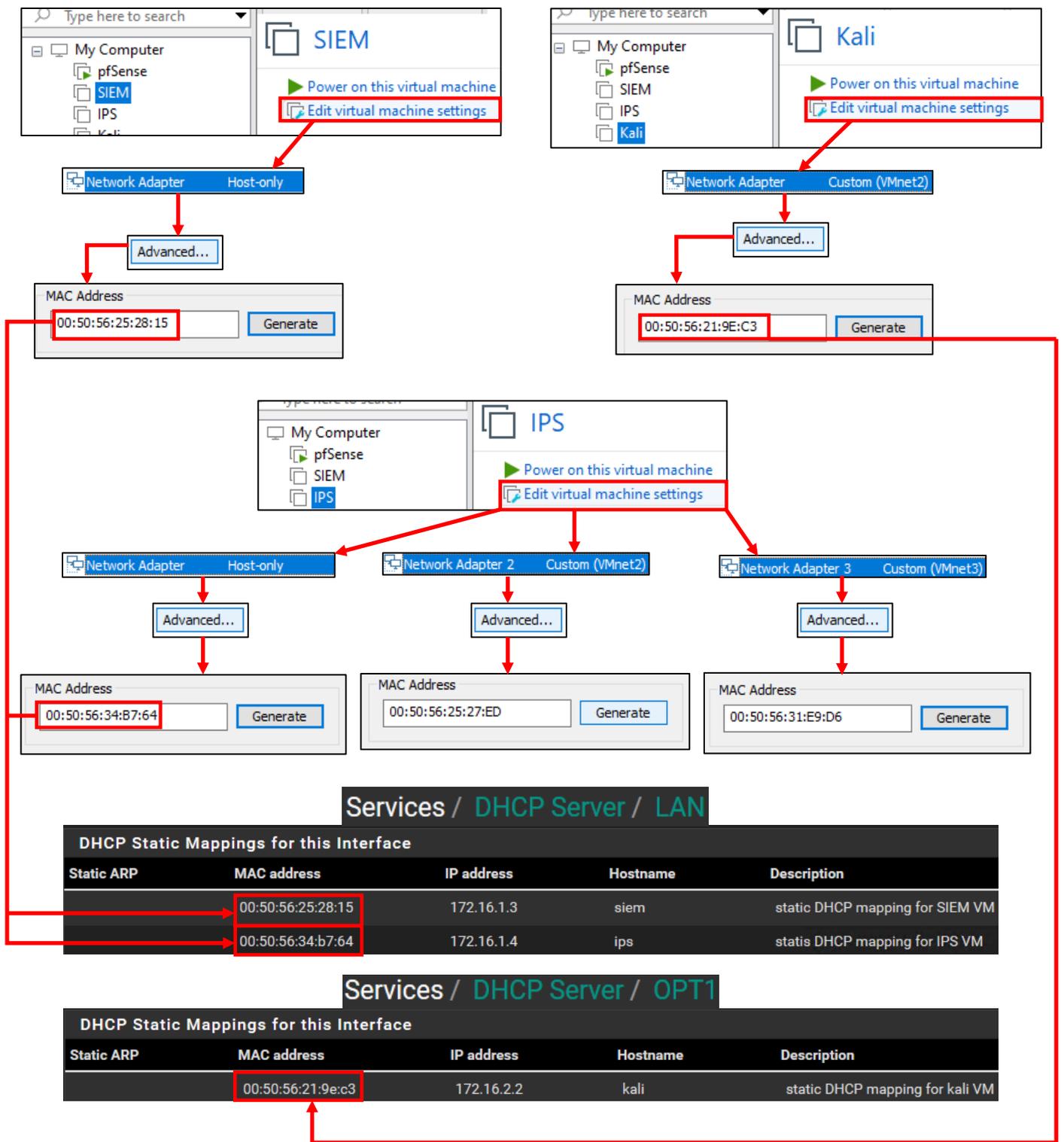
12-39: Run the *New Virtual Machine Wizard* three times, using the table provided above to create the SIEM, IPS and Kali VMs. When finished, the summary on the *Ready to Create Virtual Machine* screen should look like the summaries displayed in the illustration above.

12.6.2 Creating Static DHCP Allocations for the SIEM, IPS and Kali VMs

With all three virtual machines created, the next step will be to use the Network Adapter Advanced Settings feature and generate MAC addresses for all of the Network Adapters attached to the SIEM, IPS and Kali Virtual machines. Students will then need to document the MAC addresses generated, and use them to create static DHCP allocations on the pfSense VM.

Access the Hardware tab of the Virtual Machine Setting's menu for all three virtual machines. For a reminder on how to do this, refer to [section 12.5.3](#) (p. 441). For each virtual machine, highlight each *Network Adapter Device* listing, and click the *Advanced* button to open the *Network Adapter Advanced Settings* window. Click the *Generate* button to generate a MAC address. Record the generated MAC address, the virtual machine it belongs to and the network segment it is attached to (refer to the sidebar discussion in [section 12.5.1](#), *Noting the Notable* for a nice template students can use to document information about their virtual machines).

From there, students will log into the pfSense VM WebConfigurator, and configure static DHCP allocations for the SIEM VM, Kali VM, and the network adapter attached to the *Host-only* network of the IPS VM. The SIEM VM should be statically assigned the IP address 172.16.1.3, the host-only interface of the IPS VM should be assigned 172.16.1.4. Both of these allocations should be configured on the *LAN* interface of the IPS VM. Meanwhile, the Kali VM should be assigned the IP address 172.16.2.2 on the *OPT1* interface. Students may refer to Chapter 14, [section 14.3.4.1](#) (pp. 690-692) for a refresher on creating static DHCP mappings on pfSense.

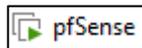


12-40: Access the *Virtual Machine Settings* menu for all three virtual machines. Using the *Advanced Network Settings* window, *Generate* and document a MAC address for each network interface on each virtual machine. Be sure to document which virtual machine and network each adapter is attached to. Log in to the pfSense WebConfigurator, and create a static DHCP mapping for the SIEM VM and the Host-only adapter of the IPS VM on the LAN interface, and a static DHCP mapping for the kali VM on the OPT1 interface. Be sure to save your changes as you proceed.

12.6.3 Operating System Installation

In this section, students will learn how to install the operating system for the SIEM, IPS, and Kali virtual machines. Both the SIEM and IPS VMs will have Ubuntu Server installed as their operating system, while the Kali VM will have the latest version of the Kali Linux distribution installed. The installation instructions will differ for each virtual machine, so please pay attention.

As a general reminder, please make sure that the pfSense VM is running, and that you have completed chapter 14 to ensure pfSense is ready to support the rest of the lab environment. Without the pfSense VM, none of the virtual machines will have internet access. That may result in the operating system installers failing in different ways. To confirm the pfSense is running, students can check tiny icon next to the name of the VM on the *Library* pane. If the VM is running, A small green arrow will appear over a set of two overlapping squares next to the VM.



12-41: The little green arrow head icon is a quick indicator that a VM is powered on and running.

12.6.3.1 Installing Ubuntu on the SIEM VM

To get started, on the *Library* pane, click the SIEM entry to highlight it, then click the text *Power on this virtual machine*. The virtual machine will begin booting off the Ubuntu Server ISO. The first screen students see will ask to confirm the language they wish to use. The default language should be *English*, so hit the enter key on your keyboard to continue.

Depending on when students downloaded their copy of the Ubuntu Server ISO, and how frequently the ISO is updated, a screen may appear titled *Installer update available*. This screen provides users with the option to download the latest version of the Ubuntu installation wizard, called Subiquity. Use the arrow keys on your keyboard to highlight *Update to the new installer*, then hit enter.

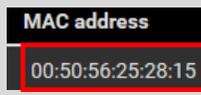
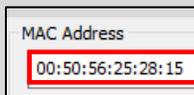
Note: If for some reason downloading the latest installer fails, there's a good chance that there are network problems with the lab environment elsewhere, and that there is troubleshooting to do. Students are welcome to select the *Continue without updating* option, but keep this in mind if the installer misbehaves or fails later. Check to see if the hypervisor host has internet connectivity, double check the firewall rules on the pfSense virtual machine, network settings, physical cabling, etc.

The next screen asks users to confirm their keyboard configuration. The default settings for both the *Layout* and *Variant* settings are *English (US)*. If you are not using a standard US-English keyboard, you may wish to use the arrow keys to highlight the *Identify keyboard* option, then hit enter. Otherwise, highlight *Done* on the bottom of the screen, and hit enter.

Next up, is the *Network connections* screen. a single network adapter should populate this page. The network adapter (named `ens32` in my case) should automatically be assigned the IP address `172.16.1.3`. Below the IP address in light grey text is the MAC address of the network adapter that the Ubuntu installer detected. This should be the same MAC address of the network adapter of the SIEM VM. If the correct IP address was assigned, students can hit the enter key to continue (The *Done* option should be highlighted by default). Otherwise, see the section *What Reservation?* for some troubleshooting tips.

What Reservation?

If for some reason the network adapter was assigned any other IP address other than `172.16.1.3`, Refer back to section *12.6.2*. Check the *MAC Address* field in the *Advanced Network Adapter Settings* window for the SIEM VM's, *Network Adapter* in its *Virtual Machine Settings* menu. Compare that MAC address to the MAC address used to create a static DHCP mapping on the *LAN* interface of the pfSense VM. Compare that to the MAC address displayed on the *Network connections* screen of the Ubuntu installer. **They should all be identical.** If there are any errors, correct them and reset the SIEM VM, to restart the Ubuntu installer until the pfSense DHCP assigns the network adapter the correct IP address.



12-42: If the SIEM VM failed to get the correct IP address, check the *Edit settings* menu of the virtual machine – specifically the *MAC Address* field under *Network Adapter 1*. Compare that to the MAC address used to create a static DHCP mapping on the *LAN* interface on the pfSense WebConfigurator. Correct the static DHCP entry as necessary then restart the SIEM VM to restart the ubuntu installer. Confirm that the network adapter was correctly assigned the `172.16.1.3` IP address.

The *Configure proxy* screen appears. Use the up arrow key to highlight the text box labeled *Proxy address* and enter `http://172.16.1.1:3128`. If you recall from Chapter 14, this is the IP address and port for the Squid proxy on the *LAN* interface of the pfSense VM. Use the arrow keys to highlight *Done*, and hit enter to continue.

The next screen, labeled *Configure Ubuntu archive mirror* will appear. This is another one of those situations where students will know whether or not they need to change this setting. Unless the lab environment is in an enterprise network and the network team happens to be operating their own software archive mirror, accept the default setting (in my case, the default mirror address was `http://us.archive.ubuntu.com/ubuntu`). With *Done* highlighted, hit enter to continue.

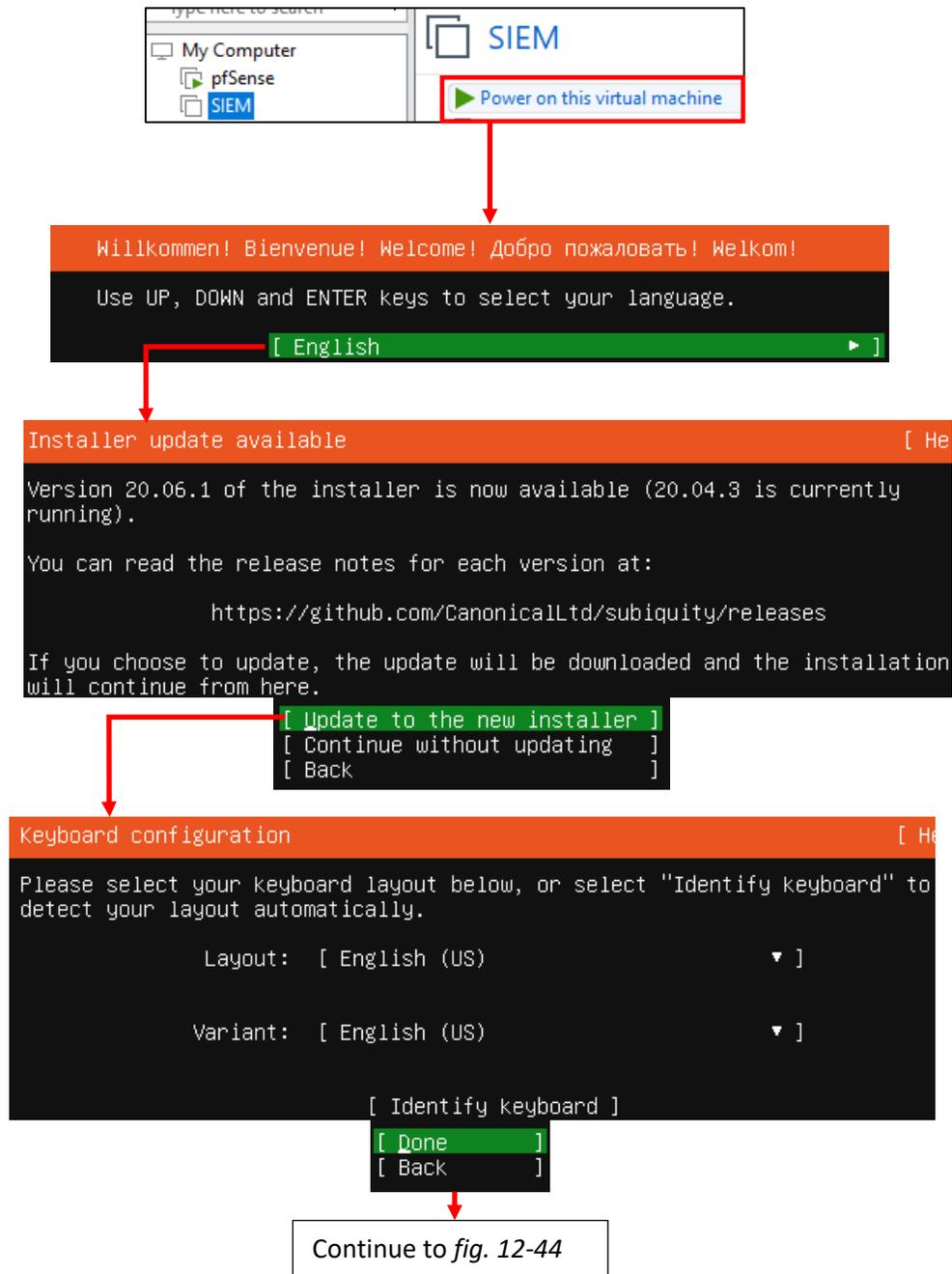
The *Guided storage configuration* screen appears next. Accept the default settings and let the Ubuntu installer format the entire disk. Use the arrow keys to highlight *Done*, and hit enter to continue. The next screen, titled *Storage configuration*, shows you how the installer is going to format the hard drive, and what partitions are going to be where in a large list labeled *FILE SYSTEM SUMMARY*. By default, *Done* should already be highlighted on this screen. If not, use the arrow keys to highlight it and hit enter to continue. A pop-up labeled *Confirm destructive action* appears. This screen informs the user that any data on the disk will be lost as a result of formatting and partitioning the disk. Since there is no data on the virtual hard disk yet, with *Continue* highlighted, hit the enter key to proceed.

Next is the *Profile setup* screen. There are five input boxes on this screen. Ubuntu asks the user for their name, the server's name, a username (that will be used to log in to the server later), the password for that username, followed by an input box asking the user to repeat the password. Students may enter any name, username, or password they would like, but it is recommended to both set the server name to *siem*, as well as to save the username and password combination to a password manager. Once finished, use the arrow keys to highlight *Done*, then hit enter to continue.

The *SSH Setup* screen appears and asks users if they would like to install the OpenSSH server package by default. By default, the prompt should be between two brackets next to the text *Install OpenSSH server*. Hit the spacebar to leave an 'X' between the brackets. Afterwards, use the arrow keys to highlight *Done* and hit enter to advance.

The next screen is labeled *Featured Server Snaps*. The latest versions of Ubuntu use an additional software manager called 'snap' to deliver software packages. Use either the arrow keys or the tab key to highlight *Done* and hit enter – Do not install any snaps, continue the installer.

We reach a new screen labeled *Install complete!* At this point, students have made all of the necessary decisions for the ubuntu installer to proceed, and handle all of the installation tasks at once. Once completed, the installer will grant students the option to reboot the system. However, instead of using the installer's reboot function, click the *Actions* option from the virtual console window, followed by *Power*, then *Power off* to shut down the virtual machine. Afterwards, close the virtual console.



12-43: In the *Library* pane, click on the SIEM VM, then click on *Power on this Virtual Machine*. This will power on the VM, and open its virtual console. The user then selects the language of the installer, the installer checks for updates for itself, then asks the user to set the keyboard layout and variant language.

Continued from *fig. 12-43*

The figure consists of three vertically stacked screenshots from the Ubuntu Server 20.04 installer, connected by red arrows indicating the flow of the installation process.

Network connections [Help]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient access for updates.

NAME	TYPE	NOTES
[ens32	eth	- ▶]
DHCPv4	172.16.1.3/24	
00:50:56:25:28:15 / Intel Corporation / 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000 MT Single Port Adapter)		

[Done]
[Back]

Configure proxy [Help]

If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user] [:pass]@]host[:port]/".

[Done]
[Back]

Configure Ubuntu archive mirror [Help]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:

You may provide an archive mirror that will be used instead of the default.

[Done]
[Back]

Continued to *fig. 12-45*

12-44: The next stages of the Ubuntu Server 20.04 installer. In these screens, students can confirm whether or not the static DHCP mapping for the SIEM VM is working correctly, configure the system to use the Squid proxy service configured on the pfSense VM, and confirm software archive mirror they would like to use.

Continued from *fig. 12-44*

```
Guided storage configuration
Configure a guided storage layout, or create a custom one:
(X) Use an entire disk
    [ /dev/sda local disk 80.000G ▾ ]
[X] Set up this disk as an LVM group
    [ Done ]
    [ Back ]
```

```
Storage configuration [ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT  SIZE  TYPE  DEVICE TYPE
[ /           39.498G new ext4 new LVM logical volume ▶ ]
[ /boot      1.000G new ext4 new partition of local disk ▶ ]

AVAILABLE DEVICES
DEVICE                                TYPE                                SIZE
[ ubuntu-vg (new)                    LVM volume group                  78.996G ▶ ]
free space                            39.498G

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES
DEVICE                                TYPE                                SIZE
[ ubuntu-vg (new)                    LVM volume group                  78.996G ▶ ]
ubuntu-lv  new, to be formatted as ext4, mounted at /  39.498G ▶ ]
[ /dev/sda                            local disk                        80.000G ▶ ]
partition 1 new, bios_grub                                       1.000M ▶ ]
partition 2 new, to be formatted as ext4, mounted at /boot  1.000G ▶ ]
partition 3 new, PV of LVM volume group ubuntu-vg        78.997G ▶ ]
```

```
[ Done ]
[ Reset ]
[ Back ]
```

Continued to *fig. 12-46*

12-45: These screens are used to configure the storage settings for the operating system. Students will be using the default storage settings for the SIEM VM.

Continued from *fig. 12-45*

```
Confirm destructive action

Selecting Continue below will begin the installation process and
result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the
installation has started.

Are you sure you want to continue?

[ No ]
[ Continue ]
```

```
Profile setup [ Help ]

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

Your name: ayy
Your server's name: siem
The name it uses when it talks to other computers.
Pick a username: ayy
Choose a password: *****
Confirm your password: *****

[ Done ]
```

Title:	SIEM VM
Username:	ayy
Password:	*****
URL:	172.16.1.3
<input type="checkbox"/> Expires:	7/21/2020 12:11 PM
<input checked="" type="checkbox"/> Notes:	User account credentials for the SIEM VM.

Continued to *fig. 12-47*

12-46: After confirming the storage configuration settings, users are prompted to name their server, and create a user account. It's recommended to store the username and password for the SIEM VM in a password manager.

Continued from *fig. 12-46*

```
SSH Setup [ Help ]
You can choose to install the OpenSSH server package to enable secure remote
access to your server.

[X] Install OpenSSH server
```

```
[ Done ]
[ Back ]
```

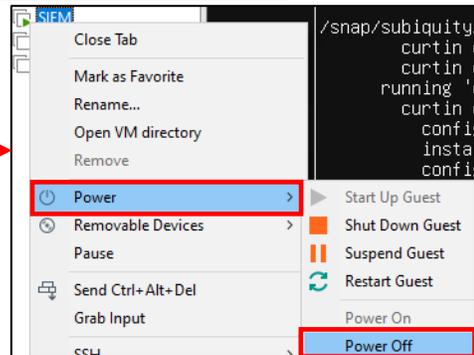
```
Featured Server Snaps [ Help ]
These are popular snaps in server environments. Select or deselect with SPACE,
press ENTER to see more details of the package, publisher and versions
available.
```

```
[ Done ]
[ Back ]
```

```
Install complete! [ Help ]

configuring installed system
  running '/snap/bin/subiquity.subiquity-configure-run'
  running '/snap/bin/subiquity.subiquity-configure-apt'
  running '/snap/subiquity/1938/usr/bin/python3 true'
  curtin command apt-config
  curtin command in-target
  running 'curtin curthooks'
  curtin command curthooks
    configuring apt
    configuring apt
    installing missing packages
    configuring iscsi service
    configuring raid (mdadm) service
    installing kernel
    setting up swap
    apply networking config
    writing etc/fstab
    configuring multipath
    updating packages on target system
    configuring pollinate user-agent on target
    updating initramfs configuration
    configuring target system bootloader
    installing grub to target devices
  finalizing installation
    running 'curtin hook'
    curtin command hook
  executing late commands
final system configuration
  configuring cloud-init
  installing openssh-server |
```

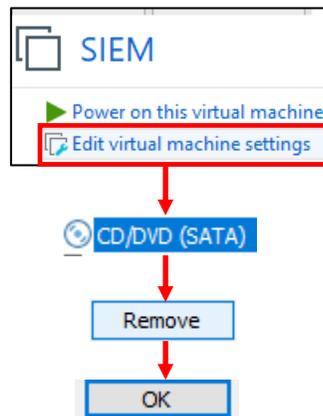
```
[ View full log ]
[ Reboot ]
```



12-47: The final stages of the Ubuntu Server 20.04 installer. Select the option to install OpenSSH server, then decline to install any server snaps. Finally, once the installer grants you the option to reboot, ***Use the Actions menu on the virtual console to power off the virtual machine.***

12.6.3.2 Additional Virtual Machine Settings – SIEM VM

Now that the operating system installation is complete, there is one last configuration setting to adjust on the SIEM VM before we can boot into Ubuntu Linux and perform some diagnostic tasks. Back in [section 12.5.3](#) (p. 441), students learned how to remove the CD/DVD Drive from the pfSense VM. Students need to perform that task on the SIEM VM. Open the SIEM VM's *Virtual Machine Settings* menu, locate the *CD/DVD (SATA) Device*, highlight it, then click *Remove*. Afterwards click the *OK* button in the lower right corner to confirm this change, and close the SIEM VM's settings menu.



12-48: Access the SIEM VM's *Virtual Machine Settings* menu and *Remove the CD/DVD (SATA) Device*, then click *OK* to exit.

12.6.3.3 Booting the SIEM VM for the first time

After changing the SIEM VM's settings, start the VM back up and bring up its virtual console. After a moment or two, you will be greeted with login prompt labeled *SIEM login*. Enter the username you configured during the installer, followed by the password to log in.

Some students may not be familiar with command-line applications, and that's okay. This is only a quick login to make sure network connectivity is working. Please type in the following commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The purpose of the `ip` command above is to display all of the network interfaces on the system. We pass this command the `-br` option for brief output, followed by the letter 'a' to indicate we're interested in seeing the IP addresses on our system. Users could replace 'a' with 'address' or 'addr' and the `ip` command would interpret it the same. We're using this command to serve as a secondary confirmation that the SIEM VM was successfully assigned the IP address 172.16.1.3,

as displayed in *fig.12-49* below. Students may notice a second interface on the system designated lo. This is a "loopback" network interface and can safely be ignored.

The nslookup command is to confirm that the SIEM VM is able to resolve hostnames using DNS. The output from the command should be similar to what is presented in *fig. 12-49*. Finally, that brings us to the curl command. This command is to confirm connectivity to the internet over port 443, HTTPS. The -I option in the command tells curl to only return the headers from the web server being contacted. Once again, the output of this command should be fairly similar to what is presented in *fig. 12-49*.

```
Ubuntu 20.04 LTS siem tty1
siem login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ nslookup www.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:806::2004
ayy@siem:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 21 Jul 2020 20:10:38 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 21 Jul 2020 20:10:38 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-21-20; expires=Thu, 20-Aug-2020 20:10:38 GMT; path=/; domain=.google.com;
Secure
set-cookie: NID=204=1AAB5nk21PEgo8rGiFr-9PxEuTIYONx2tMMi-EmACtdRnP1PkB0xoosGu9FjWzblYw0TG0HKUj6kLonn
4Rr-yu-Mic8itYAI507X2Vh2HzhKb1Wk0UK30rSK6Fd0ce6_Battd9PQ4YI7-CoRGf381n74mD78YmTAAtz1XJf8-WM; expires
=Wed, 20-Jan-2021 20:10:38 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443";
ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":4
43"; ma=2592000; v="46,43"
ayy@siem:~$ ip -br a
lo                UNKNOWN      127.0.0.1/8  ::1/128
ens32             UP           172.16.1.3/24 fe80::250:56ff:fe25:2815/64
```

12-49: After logging in to the SIEM virtual machine, students will need to run a series of network troubleshooting commands. These commands are to confirm that the SIEM VM has the correct IP address configured (`ip -br a`), can resolve hostnames via DNS (`nslookup www.google.com`), and has connectivity over HTTPS (`curl -I https://www.google.com`).

Before logging out of the SIEM virtual machine, there are three more commands to run, but before we can run them, we will need to become the root user. Enter the following command:

```
sudo su -
```

When prompted, enter the password for the user you created. If successful, you will be logged in as the root user on the SIEM virtual machine. The root user, sometimes referred to as the super user, is a special account that has complete authority over the system. Additionally, root has access to special administrative commands that normal users are not allowed to use. As the root user, let's **run those last three commands in this exact order:**

```
apt-get update
apt-get -y dist-upgrade
init 6
```

Ubuntu is based off of the Debian Linux distribution. Because of this, it uses a package manager called apt (in addition to the snap package manager mentioned earlier). The two apt-get commands, apt-get update then apt-get -y dist-upgrade tell Ubuntu to reach out to the software archive mirror and get an updated list of software packages, then if any packages installed on the system need to be updated, updated them immediately. This set of commands also confirms that the Squid proxy server on the pfSense VM is working properly, and proxying all of the HTTP requests from the SIEM VM. The final command, init 6, tells the system to reboot immediately. As an alternative, users can also run the command reboot instead.

```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (524 kB/s)
Reading package lists... Done
root@siem:~# apt-get -y dist-upgrade
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@siem:~# init 6_
```

12-50: the command `sudo su -` allows students to become the root user on the SIEM VM. We then use root's permissions to ensure all of the software packages on the system are up to date (`apt-get update`, followed by `apt-get -y dist-upgrade`), then immediately reboot the system (`init 6`, or optionally `reboot`). Be aware that the apt-get commands may take a little bit of time to finish, based on the number of updates available and speed of your internet connection.

Help! My apt-get commands are failing!

If you're experiencing problems with the apt-get commands failing to complete, there's a very good chance that the apt package manager is not properly configured to use the SQUID HTTP proxy we installed on the pfSense VM, or that the SQUID proxy service on the pfSense VM may be misconfigured. If students entered the wrong information during the operation system installation (e.g., on the *Configure Proxy* screen), then the apt package manager will not work properly.

Here are some troubleshooting steps to think about:

On the SIEM VM, run the command:

```
cat /etc/apt/apt.conf.d/90curtin-aptproxy
```

This command will read the contents of the file `/etc/apt/apt.conf.d/90curtin-aptproxy` and display its contents on the screen. The file should read something like this:

```
Acquire::http::Proxy "http://172.16.1.1:3128";  
Acquire::https::Proxy "http://172.16.1.1:3128";
```

If this file does not exist, or has any content that is in any way different from the lines above, **run the following three commands exactly as displayed, and in this exact order:**

```
sudo su -  
echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-  
aptproxy  
echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >>  
/etc/apt/apt.conf.d/90curtin-aptproxy
```

This series of commands requires root access, so the first thing we do is use `sudo su -` to become the root user. The next two commands delete the current `90-curtin-aptproxy` file if it exists, then overwrites it with the two correct entries that should exist in the file. After running these commands, run `cat /etc/apt/apt.conf.d/90curtin-aptproxy` once more, and confirm that the output matches the correct output listed above. After confirming that the configuration file has been recreated correctly, try running the apt-get commands once more. If they continue to fail, then continue the troubleshooting process. Assuming that the network connectivity check commands were successful (e.g., `nslookup` and `curl`), think about the following:

- Is the SQUID proxy service installed on pfSense?
- Is there a firewall rule on the LAN interface to allow access to the proxy service? (allow traffic to IP address 172.16.1.1 port 3128 TCP from network 172.16.1.0/24)
- Is the option *Resolve DNS IPv4 First* checked on the SQUID proxy service?

These are all configurations covered in chapter 14, and should have already been specified. Double check that they have been configured correctly, then try updating the SIEM VM again.

```

ayy@siem:~$ 1 sudo su -
root@siem:~# 2 echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 3 echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >> /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 4 cat /etc/apt/apt.conf.d/90curtin-aptproxy
Acquire::http::Proxy "http://172.16.1.1:3128";
Acquire::https::Proxy "http://172.16.1.1:3128";
root@siem:~# 5 apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [332 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [8,780 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [5,404 B]
Fetched 826 kB in 1s (972 kB/s)
Reading package lists... Done
root@siem:~# 6 apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@siem:~#

```

12-51: This illustration demonstrates how to fix or modify the `/etc/apt/apt.conf.d/90curtin-aptproxy` file, in the event that student find that there is a problem with the file. First utilize `sudo su -` (1) to become the root user. Then use the two `echo` commands (2, 3) to write the correct configuration data so that `apt` knows how and where to access the squid proxy configured on the pfSense VM. Utilize the `cat` (4) command to confirm that the configuration file is properly configured. Finally, run `apt-get update` (5) and `apt-get -y dist-upgrade` (6) to check for the latest updates and download them.

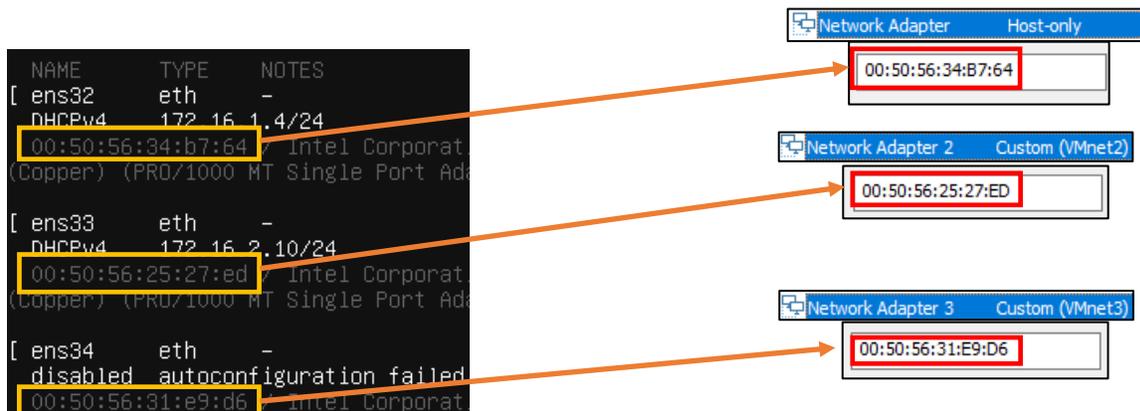
12.6.3.4 Installing Ubuntu on the IPS VM

Now that Ubuntu has been installed on the SIEM VM, network connectivity has been checked, and updates have been applied, next up is the IPS VM. The process for installing Ubuntu Server on the IPS virtual machine is practically identical, the process will be summarized below, with major differences to be aware of explained further in-depth.

- Start the IPS VM, and connect to its virtual console
- Select English as your language (or your preferred language)
- If there are any updates to Subiquity, select the option, *Update to the new installer*
- Select *English (US)* (or your preferred language) as the keyboard *Layout* and *Variant*

The *Network connections* screen will be a little bit different than it was on the SIEM VM, because the IPS virtual machine has three network interfaces. Recall in [section 12.5.4.1](#) (pp. 442-445), comparing and contrasting the MAC addresses of the three network adapters attached to the pfSense VM, and using that information to correctly map the network interfaces to the correct networks.

Students will need to perform a similar exercise for the IPS virtual machine on the *Network connections* screen. In light grey text underneath the name of each network interface is the MAC address for that interface. Cross-reference the MAC address and interface name on the screen with the MAC address of network adapters 1-3 recorded earlier. See *fig. 12-52* below for an example, based on the MAC addresses of my IPS virtual machine.



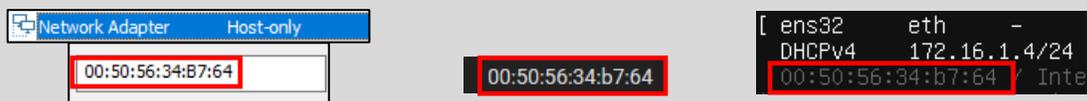
12-52: Ubuntu has assigned our three virtual adapters the interface names *ens32*, *ens33*, and *ens34*. Below each interface name is a MAC address. By cross referencing those MAC addresses to the MAC addresses and the virtual switches attached to the IPS VM, we can determine which interface name maps to which network adapter, and confirm which virtual switch the interfaces are attached to. For instance, *ens32*, is the network adapter attached to the *Host-only* network. *ens33* is attached to the *VMnet2 (IPS1)* network, while *ens34* is attached to the *VMnet3 (IPS2)* network.

Now that students are aware of which interface corresponds to which network segment, the next step is ensuring that the interface connected to the *Host-only* (e.g., the LAN network in pfSense) is the only interface that has an IP address assigned. In section 12.6.2, *fig. 12-40*, students created a static DHCP reservation for the IPS VM using the MAC address of the network adapter attached to the *Host-only* network, and assigned it the IP address 172.16.1.4. In *fig. 12-52* above, the interface `ens32` has the IP address 172.16.1.4. This confirms students created the static DHCP allocation correctly on the pfSense WebConfigurator, and that `ens32` is the interface connected to the *Host-only* network. If the network adapter attached to the LAN network does not have the correct IP address, there is a good chance that the static DHCP mapping for the IPS virtual machine is incorrect. Take a look at the side bar discussion, *Reservation for One*, for some troubleshooting recommendations.

Reservation for One

If you're here, that means that the network interface attached to the LAN/Management network didn't get the IP address 172.16.1.4. Similar to the *What Reservation?* Sidebar discussion for the SIEM VM, you'll want to check a few things:

- Check the MAC address of the network adapter attached to the *Host-only* network
- Visit *Services > DHCP Server* and Check the Static DHCP Mappings of the *LAN* interface, particularly, the entry for the IPS VM
- Compare the MAC address of the previous two locations with the MAC addresses presented on the *Network Connections* screen. You should already know which interface name maps to which MAC address and network segment. In my case this was the interface `ens32`
- Make any corrections to the static DHCP allocation, then restart the IPS VM. Make your way back to the *Network connections* screen, and confirm that the correct interface was assigned the correct IP address.



12-53: Just like with the SIEM VM, compare the MAC address of the network adapter attached to the *Host-only* network. Compare that to the MAC address used to create a static DHCP mapping for the IPS VM on the LAN interface of the pfSense webConfigurator. If they don't match, correct the static DHCP mapping entry, then restart the IPS VM. Determine which interface was assigned the IP address 172.16.1.4.

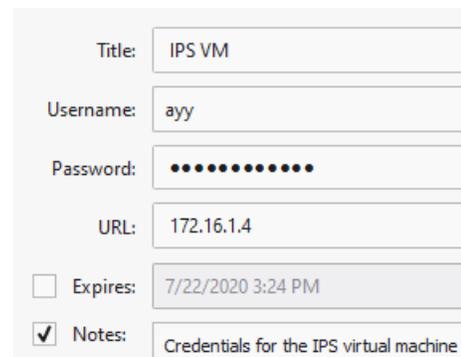
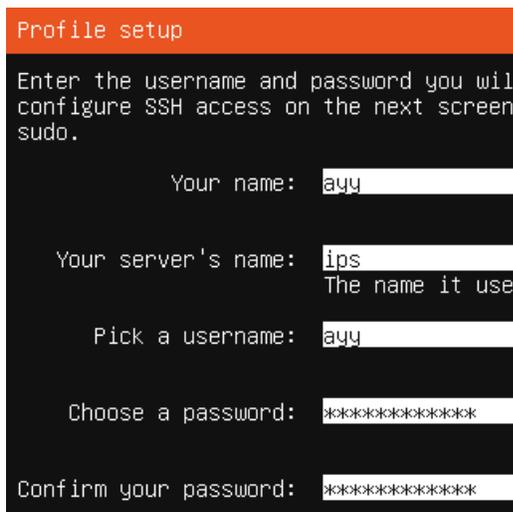
The final step on the *Network connections* screen is to disable the remaining network interfaces. The interfaces connected to the IPS1 and IPS2 port groups (In the illustrations provided, these are the interfaces *ens33* and *ens34*) should never receive an IP address. The lab environment, and IPS software we'll be using does not require these interfaces to have IP addresses, so we want to take advantage of that. Students may have notice that the interface connected to the VMnet3 (*IPS2*) network (*ens34*) doesn't have an IP address assigned, instead displaying the status: *disabled autoconfiguration failed*. Disregard this error message and follow the instructions below. Substitute the interface names *ens33* and *ens34* as necessary:

- Using the arrow keys, Highlight one of the other remaining interfaces. In my case, I chose to highlight *ens33*. Hit enter, and a dialogue box pops up.
 - Highlight the option *Edit IPv4*, and hit enter.
 - A new dialogue box appears titled *Edit ens33 IPv4 configuration*, with a single drop-down option highlighted, titled *IPv4 Method*. Hit enter again, and a list of choices appear. Use the arrow keys to select the option *Disabled*, and hit enter.
 - Use the arrow keys to highlight the option *Save*, and hit enter.
- Optional: Repeat the process again, only this time, Select *Edit IPv6*. By default, IPv6 should already be set to disabled, so this should not be necessary, but it is important to ensure these interfaces never receive an IPv4 or IPv6 address.
 - When finished, exit the *Edit ens33 IPv6 configuration* dialogue box.
- Repeat this process for the final interface. In my case, *ens34*. Disable the IPv4 Configuration (in my case, it was already set to *Disabled*) and confirm that the IPv6 configuration is already *Disabled*.

The end result should be one interface with the IP address 172.16.1.4, and two disabled network interfaces. Students can refer to *fig. 12-54* below for assistance. When finished, use the arrow keys to highlight *Done*, and hit enter to continue.

The rest of the installation process for the IPS VM should be identical to the SIEM VM:

- On the *Configure proxy* screen, set the *Proxy address* to `http://172.16.1.1:3128`
- Accept the default archive mirror (or an alternative, if required) on the *Configure Ubuntu archive mirror* screen
- Accept the default settings on the *Guided storage configuration*, and *Storage configuration* screens. Select *Continue* on the *Confirm destructive action* dialogue pop-up
- Fill out the *Profile setup* screen, ensuring that the *Your server's name* input box is set to *ips*. Remember to document the username and password you create and store it in your preferred password manager

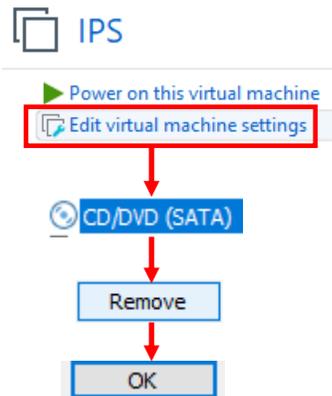


12-55: The *Profile setup* screen for the IPS virtual machine is, quite literally, the only other screen aside from the *Network connections* screen that differs from the installation process used on the SIEM VM. As always, save your username and password to a password manager!

- On the *SSH Setup* screen, be sure to select *Install OpenSSH server*
- On the *Featured Server Snaps* screen, select *Done* and hit enter to move on to the *Installation complete* phase
- Once the installation has finished, use the *Power off* option from the virtual console's *Actions* menu to shut down the virtual machine, just like the SIEM VM.

12.6.3.5 Additional Virtual Machine Settings – IPS VM

Now that Ubuntu Server is installed on the IPS VM, all that is left is to remove the *CD/DVD Drive (SATA)* virtual hardware. The process is identical to the one used on the SIEM virtual machine – open the IPS VM's *Virtual Machine Settings* menu, locate the *CD/DVD (SATA) Device*, highlight it, then click *Remove*. Afterwards click the *OK* button in the lower right corner to confirm this change, and close the IPS VM's settings menu.



12-56: Access the IPS VM's *Virtual Machine Settings* menu and *Remove the CD/DVD (SATA) Device*, then click *OK* to exit.

12.6.3.6 Booting the IPS VM for the first time

Power on the IPS VM, connect to its virtual console, and once Ubuntu has finished starting up and performing its first-time boot routines, log in with the username and password assigned on the *Profile setup* screen during the install. Just like with the SIEM VM, students will run the following three commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output from the `ip -br a` command will differ slightly, because the IPS VM has more network interfaces than the SIEM VM, but aside from that, the output from `nslookup` and `curl` should be more or less identical to the output of these commands from the SIEM VM. See fig. 12-57 below for an example on what the output of these commands should look like.

```

Ubuntu 20.04 LTS ips tty1

ips login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@ips:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
ens32             UP                172.16.1.4/24 fe80::250:56ff:fe34:b764/64
ens33             DOWN
ens34             DOWN
ayy@ips:~$ nslookup www.google.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.8.196
Name:   www.google.com
Address: 2607:f8b0:4009:815::2004
ayy@ips:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Thu, 23 Jul 2020 17:21:51 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Thu, 23 Jul 2020 17:21:51 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-23-17; expires=Sat, 22-Aug-2020 17:21:51 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=No2uEqnF70q9zD3pzs0rY1b9m4o1HDDzP4BzZ1ULDM2ia7uXqWv97cWdZN0fc2JxijI_BXyxhRfuF2EEvFV50ssKkaJRIZPxm4TbiDfzAihP6aW6FsTqHu6Kif6j75q06iuFFU-UP0oA73r0ytPyD314nvxBKnu1_rqEmla0Sic; expires=Fri, 22-Jan-2021 17:21:51 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"

```

12-57: Just like with the SIEM VM, students will log in to the IPS virtual machine and run a couple of network diagnostic commands. The output from `curl` and `nslookup` commands should be more or less identical to the output on the SIEM VM, but the `ip -br a` command will produce a few more lines of content. Ignoring the `lo` (loopback) interface, there should be three interfaces. Only one of them should have the status of UP. That interface should be the interface assigned to the LAN/Host-only network, with the IP address 172.16.1.4.

After running these commands to confirm the IPS VM has been assigned the proper IP address, can resolve hostnames, and has HTTPS connectivity, run the commands:

```

sudo su -
apt-get update
apt-get -y dist-upgrade
init 6

```

In order to become the root user, install updates on the IPS VM (and confirm the Squid proxy server is proxying the IPS VM's HTTP requests), then reboot after the system is done installing those updates.

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [306 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [114 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [7612 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [136 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [4792 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Fetched 889 kB in 1s (1269 kB/s)
Reading package lists... Done
root@ips:~# apt-get -y dist-upgrade
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@ips:~# init 6

```

12-58: These commands are identical to the ones students ran on the SIEM virtual machine, and the serve the same purpose for the IPS VM: become the root user, check for updated packages, install those updates, then reboot the system.

Note: If you're having problems with your apt-get commands failing, refer back to the sidebar conversation on pp. 475-476, [*Help! My apt-get commands are failing!*](#) For further guidance. Students can follow the exact same steps laid out for the SIEM VM to troubleshoot the problem.

12.6.3.7 Installing Kali Linux on the kali VM

Now that the SIEM and IPS virtual machines are out of the way, next up is the kali VM. Power on the VM, then Connect to its virtual console. A boot menu appears with a number of options. Using the arrow keys, highlight *Install* and hit enter.

Similar to the Ubuntu installer, the first screen, titled *Select a language*, asks users to choose the language they want to use for their installation. The default setting is *English*, use the arrow keys to highlight another language as necessary, then hit enter. The next screen, *Select your location*, asks users to choose what country, territory or area in which they are located. This screen defaults to *United States*. Use the arrow keys to change this value as necessary, and hit enter to continue. Next up is the *Configure the keyboard* screen, that asks the user what keymap to use for their installation. The default setting is *American English* and can be changed with the arrow keys. After highlighting a keymap, hit enter to continue.

The installer begins loading other phases and components it will need later. Afterwards, it will attempt to get an IP address. The pfSense DHCP server should give it an IP address through the OPT1 DHCP server, but students will not be able to confirm if the IP address 172.16.2.2 was

correctly assigned until after the operating system is installed. The next screen, titled *Configure the network*, prompts users to enter a hostname for the system. Students should use the default hostname *kali*. Hit the enter key to continue to the next screen that prompts for a domain name. Again, students may hit enter and accept the default, *localdomain*.

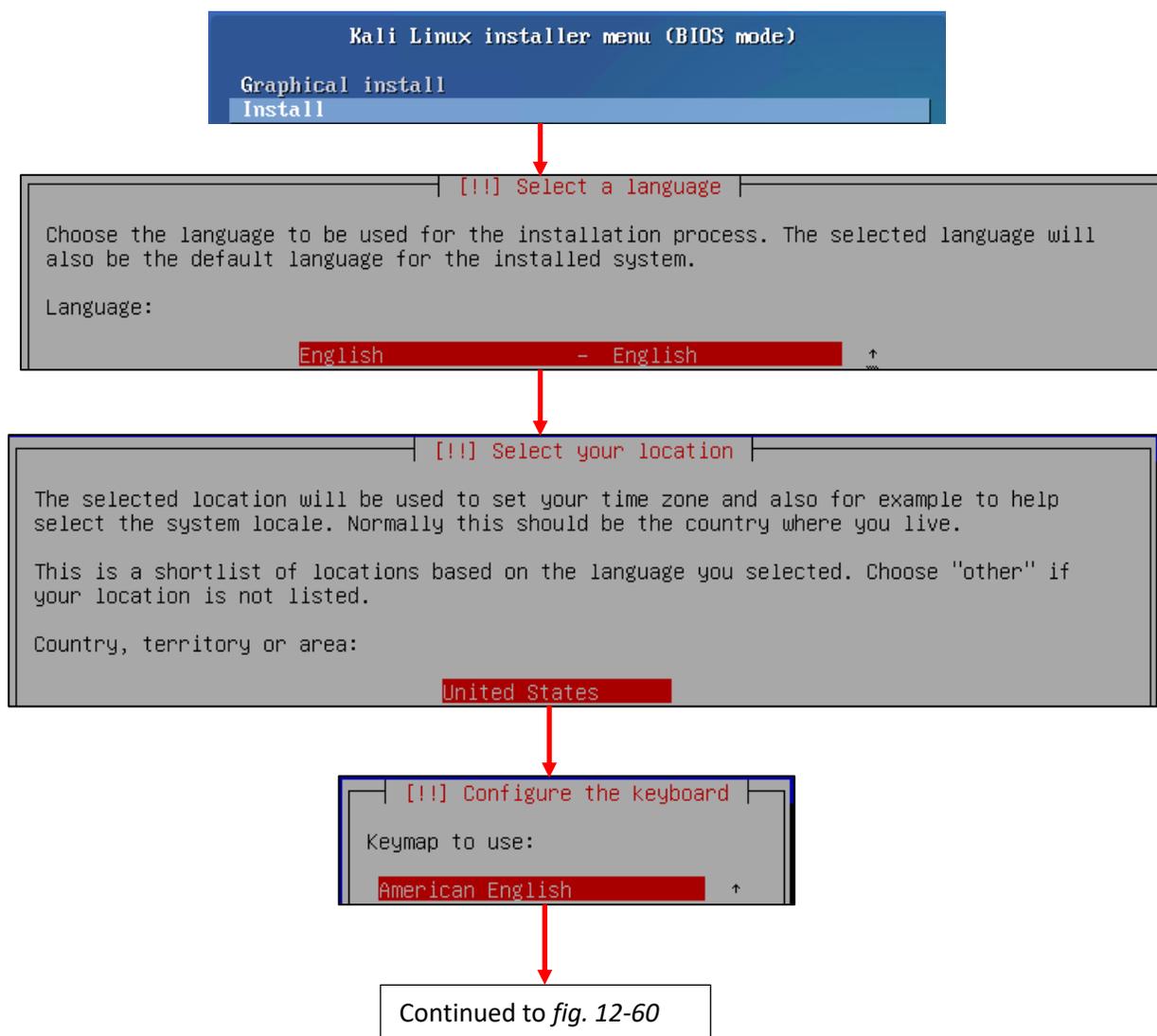
The *Set up users and passwords* screen appears. The first window asks for the full name of the user to be created. Type in the full name of the user account, and hit enter to continue to the next screen, that prompts for a username students will use to log in to the system. After typing in a username, hit enter to be prompted to create a password for this account. After hitting enter again, you'll be prompted to enter the same password again to confirm your choice. Enter the same password and hit enter to continue the installer. Just like with the SIEM and IPS virtual machines, be sure to save the username and password to your preferred password manager.

Next up is the *Configure the clock* dialogue. The installer will reach out to its preferred NTP servers to get the current time, then ask the user to select the time zone in which they are located. Use the arrow keys to choose your time zone, and hit enter to continue.

The *Partition disks* screen appears and asks users to select a partitioning method. Highlight the selection *Guided – use entire disk*, and hit enter. Users are then prompted to select the disk to partition. Since there is only a single virtual disk for the kali VM, hit enter to proceed. The next screen prompts students to select the partitioning scheme. Highlight the option *All files in one partition (recommended for new users)* and hit enter. Users are asked to confirm their choices on the next screen. Highlight the option *Finish partitioning and write changes to disk*, and hit enter. One final pop-up appears to annoy you, asking if students are sure they want to proceed, highlight *<Yes>* to confirm your choices, and press enter to continue.

The installer proceeds and begins installing the base operating systems components to the newly partitioned disk. After a moment or two, a window labeled *Software selection* appears. As the name implies, this screen allows users to pick additional software packages to install. Accept the default selections by pressing the tab key to highlight *<Continue>*, and hitting enter the next portion of the installer Retrieves and installs the requested packages. This portion of the installation may take some time, depending on internet speed and virtual machine performance.

After some time has passed a new prompt appears, labeled *Install the GRUB boot loader on a hard disk*, asking if users want to install the GRUB boot loader. This is a necessary component in order to boot the virtual machine, so highlight *<Yes>*, and press the enter key to continue. The next screen asks what partition to install the boot loader to. Seeing as how there is only one partition available, highlight it, and hit enter to proceed. After a moment or two passes, students are prompted to remove the installation media, and reboot the virtual machine to complete the installation. Just like with the SIEM and IPS virtual machines, *Turn Off* the virtual machine, then close the virtual console.



12-59: The first screens have users select their preferred language, location, and keyboard keymap.

Continued from *fig. 12-59*

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

kali

<Go Back> <Continue>

Domain name:

localdomain

<Go Back> <Continue>

Continued to *fig. 12-61*

12-60: The next few screens configure some of the network settings. Students are prompted to enter a hostname and domain name. Students should use the default hostname of *kali*, and default domain name of *localdomain*.

Continued from *fig. 12-60*

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

ayy lmao

<Go Back> <Continue>

Username for your account:

ayy

<Go Back> <Continue>

Choose a password for the new user:

[] Show Password in Clear

<Go Back> <Continue>

Re-enter password to verify:

[] Show Password in Clear

<Go Back> <Continue>

Title:	kali VM
Username:	ayy
Password:	*****
URL:	172.16.2.2
<input type="checkbox"/> Expires:	7/24/2020 11:35 AM
<input checked="" type="checkbox"/> Notes:	credentials for the kali VM

Continued to *fig. 12-62*

12-61: Similar to the *Profile Setup* screen in the Ubuntu installer, the Kali Linux installer features a series of prompts to create a user account for the system. Be sure to save the credentials to your preferred password manager when finished.

Continued from *fig. 12-61*

[!] **Configure the clock**

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

Eastern

[!!] **Partition disks**

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk

Select disk to partition:

SCSI3 (0,0,0) (sda) - 85.9 GB VMware, VMware Virtual S

Partitioning scheme:

All files in one partition (recommended for new users)

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI3 (0,0,0) (sda) - 85.9 GB VMware, VMware Virtual S

#1	primary	84.9 GB	f	ext4	/
#5	logical	1.0 GB	f	swap	swap

Undo changes to partitions
Finish partitioning and write changes to disk

Write the changes to disks?

<Yes> <No>

Continued to *fig. 12-63*

12-62: After setting the time zone, students will have to configure the partitioning scheme for the install. The highlighted options above should be selected by default. If not, use the arrows to select them, and press enter to continue.

Continued from *fig. 12-62*

```
[!] Software selection

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

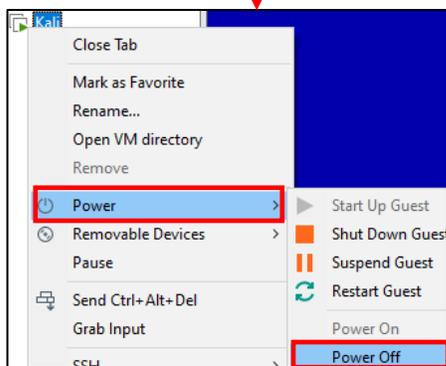
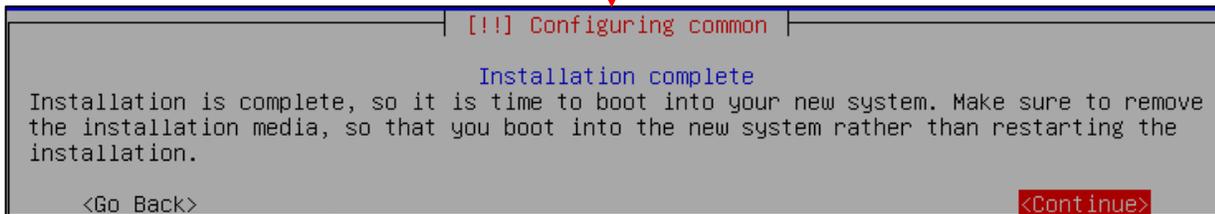
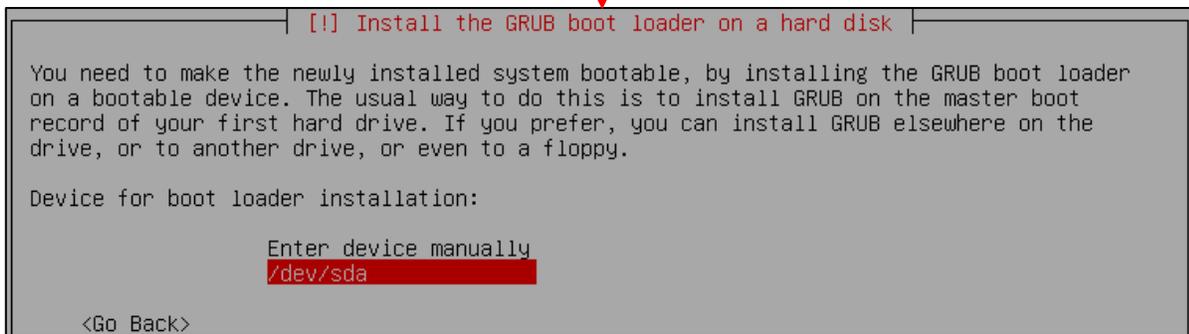
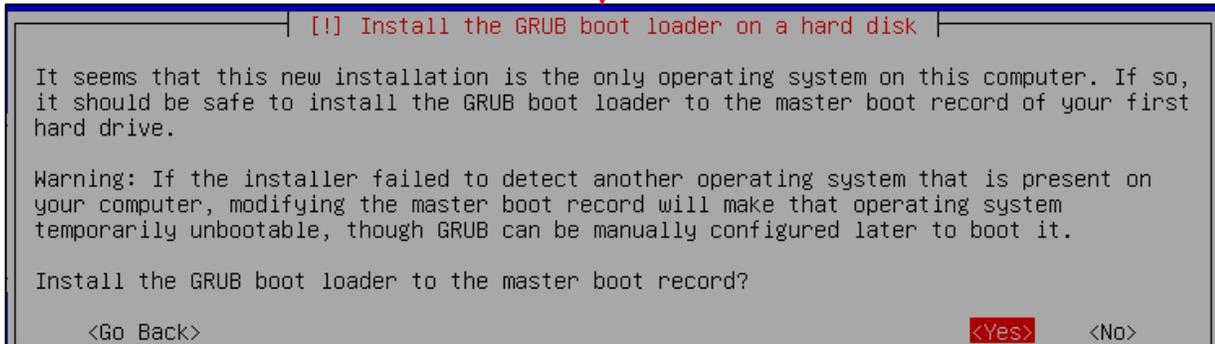
[*] Desktop environment [selecting this item has no effect]
[*] ... Xfce (Kali's default desktop environment)
[ ] ... GNOME
[ ] ... KDE Plasma
[*] Collection of tools [selecting this item has no effect]
[*] ... top10 -- the 10 most popular tools
[*] ... default -- recommended tools (available in the live system)
[ ] ... large -- default selection plus additional tools

<Continue>
```

Continued to *fig. 12-64*

12-63: On the *Software selection* screen, press the tab key to highlight *<Continue>*, and accept the default packages.

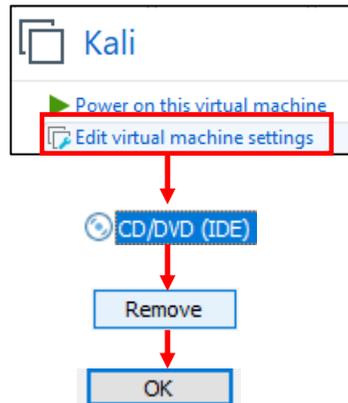
Continued from *fig. 12-63*



12-64: The final steps of the installation process. Install the GRUB boot loader to the only available disk on the system (this should be `/dev/sda`), wait for the screen labeled *Configuring common* to appear, then *Power Off* the virtual machine.

12.6.3.8 Additional Virtual Machine Settings – kali VM

By this point, it should already be established routine that once students are finished installing the operating system on their virtual machine, the next step is to remove the CD/DVD Drive, and Kali is no exception. Access the *Virtual Machine Settings* menu for the kali VM and remove the CD/DVD (IDE) device.



12-65: Access the Kali VM's *Virtual Machine Settings* menu and *Remove the CD/DVD (IDE) Device*, then click *OK* to exit.

12.6.3.9 Booting the kali VM for the first time

With those last-minute virtual machine settings applied, *Start* the Kali virtual machine, then *Connect* to its virtual console. After a moment or two passes, students will be greeted with a graphical interface, asking for a username and password to log in. Enter the username and password supplied during the operating system install, and click *Log In* to continue.

On the top of the graphical user interface, there should be a menu bar with a few icons displayed. One of those icons is a small black window. Click on that icon to open a terminal session on the kali VM. With the terminal window open, run the same three commands that we ran on the SIEM and IPS virtual machines in order to confirm network connectivity is working as intended:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output of `ip -br a` should confirm that only a single interface (again, ignoring the `lo` interface) is installed on the system. That interface should have the IP address 172.16.2.2. As with the SIEM and IPS virtual machines, if this is not the case, students should compare the MAC address of network adapter on the kali VM to the MAC address of the static DHCP mapping made on pfSense. **Make sure the mac addresses match, and that the mapping was created on the OPT1 interface.**

As with the SIEM and IPS VMs, nslookup confirms the ability of the kali VM to resolve hostnames through DNS, and the curl command verifies that the VM can make outbound internet connections over HTTPS. The output of these commands should be similar to the output displayed in *fig. 12-66* below.

While Kali Linux is slightly different from Ubuntu, we can still use *most* of the same commands utilized on the SIEM and IPS virtual machines to become root, check for updates, then reboot the system. **Run these commands in this exact order:**

```
sudo su -
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
apt-get update
apt-get -y dist-upgrade
init 6
```

Students may have noticed two new commands have been added here:

```
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
```

These commands are responsible configuring the apt package manager to use our HTTP proxy at 172.16.2.1:3128 on the *OPT1* interface of the pfSense VM. This is done by running the echo command, and redirecting its output (the > symbol) to the file /etc/apt/apt.conf.d/99local (a configuration file that the package manager will read when we run apt-get later). The second command, cat /etc/apt/apt.conf.d/99local, reads the contents of the file. If the output from the cat command reads:

```
Acquire::http::Proxy "http://172.16.2.1:3128";
```

Then that means apt was successfully configured to use the HTTP proxy. If the output from the cat command displays anything else, then students should re-enter the echo command.

Note: If most of these commands look familiar, it's because they're very similar to the troubleshooting commands I recommended in the sidebar discussion [*Help! My apt-get commands are failing!*](#) (pp. 475-476) for the SIEM and IPS virtual machines. There are a few key differences with the kali VM to be aware of, but for the most part, the troubleshooting steps laid out are the same as the steps I laid out in this section. Here are the key differences to be aware of:

- Make absolutely sure you are redirecting the output of the echo command to the file /etc/apt/apt.conf.d/99local. **It must be that exact file, in that exact location.**
- The kali VM doesn't need the second line, Acquire::https::Proxy "http://172.16.2.1:3128";
- Make absolutely sure to specify http://172.16.2.1:3128 as the proxy address for the kali VM.

After running these commands to configure the package manager, students should be able to run the remaining commands just like on the SIEM and IPS virtual machines. Bear in mind that Kali Linux is subject to frequent updates, and that some of those updates can be quite large. This means that depending on the performance of the Kali VM, and internet connection speeds, downloading and installing updates may take some time to complete.

```

ayy@kali:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
eth0              UP                172.16.2.2/24 fe80::581c:6ed2:259e:5cb/64
ayy@kali:~$ nslookup www.google.com
Server:          172.16.2.1
Address:         172.16.2.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:804::2004
ayy@kali:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 24 Jul 2020 19:55:47 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 24 Jul 2020 19:55:47 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-24-19; expires=Sun, 23-Aug-2020 19:55:47 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=rC1q-094PKdmAIZC2ajgCkpdGrGdulzdaxnJR2Cui-HYKBgbjzh_qCz6G5tJYoetE_Uc7rscR53x4Gri7HE3k_gu9h2BKh6etyF0hGD0iat3FF22oe-4VngjLFAdGEY3XTecVHB8iJ5Qw2quVjmmN7oZGeRUSj1mXJ8ulaLkRY; expires=Sat, 23-Jan-2021 19:55:47 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@kali:~$ sudo su -

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for ayy:
root@kali:~# echo 'Acquire::http::proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
root@kali:~# cat /etc/apt/apt.conf.d/99local
Acquire::http::proxy "http://172.16.2.1:3128";
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@kali:~# init 6

```

12-66: Login to the kali VM, configure the apt package manager to use the SQUID HTTP proxy on *OPT1* of the pfSense VM. Afterwards, install the latest operating system updates, then reboot the virtual machine.

12.6.4 Metasploitable 2

The Metasploitable 2 Virtual is covered separately from the other VMs students have created because it's a special case. Technically the VM is already created, all that needs to be done is to "register" with VMware Workstation. However, there are a number of small configuration tasks that need to be covered both before and after registering the virtual machine.

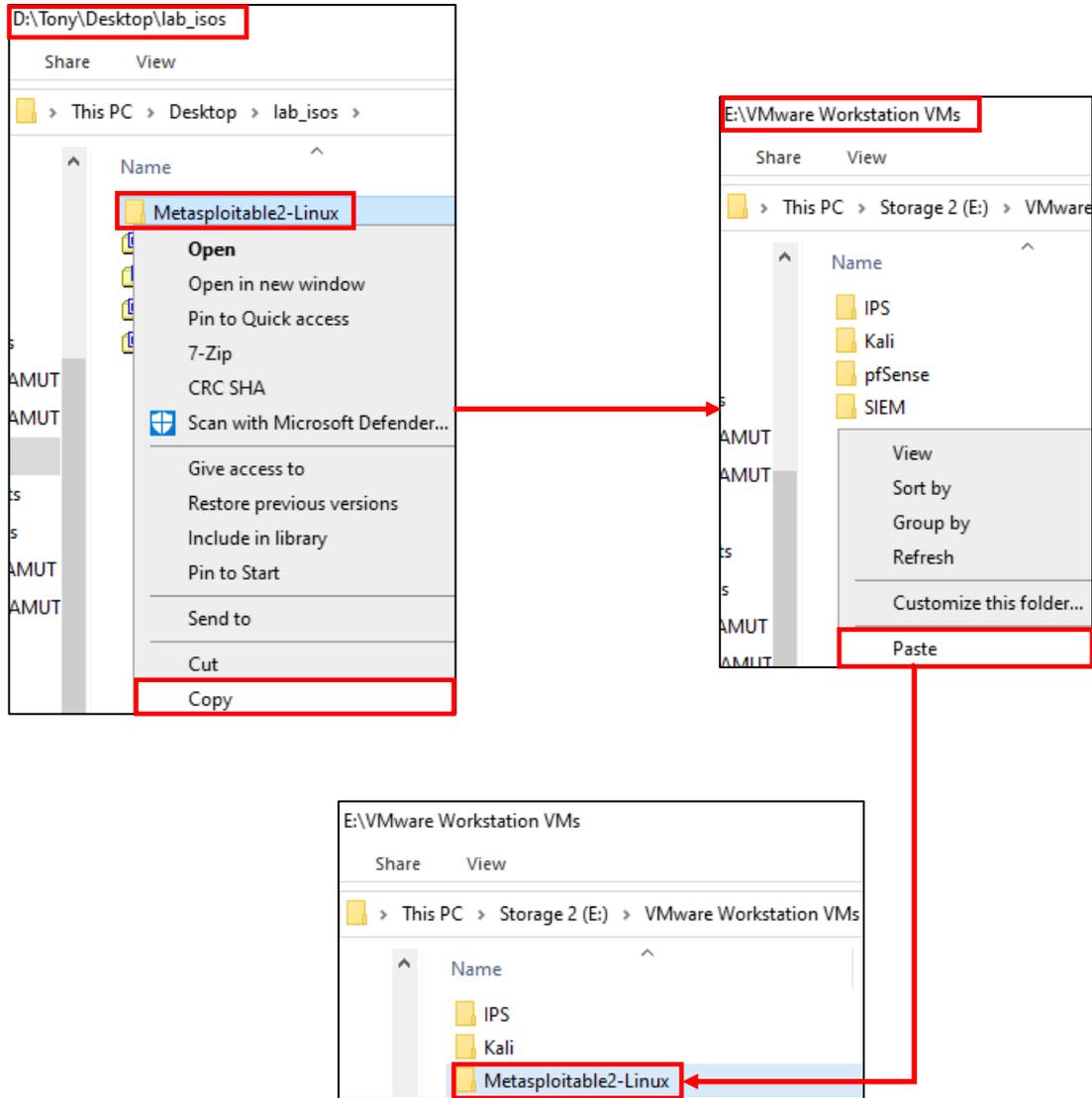
12.6.4.1 Registering the Metasploitable 2 VM

Note: Before we begin, please make sure that you have downloaded the Metasploitable 2 VM from Sourceforge, and that you've decompressed the `metasploitable-linux-2.0.0.zip` file. For guidance, check out Chapter 1, [section 1.8](#) (*Using Compression Tools*, pp. 33-35). you'll need access to the entire `Metasploitable2-Linux` directory to perform the tasks ahead.

Back in [section 12.2](#) (pp. 413-416), we discussed the *Preferences* setting *Default location for virtual machines*. The first task students will need to perform is copying the entire `Metasploitable2-Linux` directory to that configured directory.

Windows Users: Use the windows explorer file browser, and browse to the directory where the decompressed `Metasploitable2-Linux` directory is located. For example, on my computer I stored all of the virtual machine ISOs and the `Metasploitable2-Linux` directory in a directory on my desktop labeled `lab_isos`. The full path is `D:\Tony\Desktop\lab_isos`. Right-click on the `Metasploitable2-Linux` directory, and select *Copy*.

Next, using the file browser, navigate to the Default location for virtual machines directory. Again, on my system, I configured this to be `E:\VMware Worktation VMs`. Students will know they are in the right place if they see folders named after the other virtual machines they have already created – `pfSense`, `SIEM`, `IPS`, and `kali`. Right click on a blank space in the folder, and select the *Paste* option to copy the entire `Metasploitable2-Linux` directory into this folder.



12-67: Locate the decompressed Metasploitable2-Linux directory. On my system, I stored it in `D:\Tony\Desktop\lab_isos`. Right click on it and select `Copy`. Navigate to the Default location for virtual machines configured for the VMware Workstation install. As a reminder, students can access the Preferences menu (`Edit > Preferences`), and the Workspace setting allows students document and/or modify the default directory used for storing their virtual machines. Once again, on my system, I changed this to `E:\VMware Workstation VMs`. Navigate to the Default location for virtual machines directory, right click within it, and select `Paste` to copy the entire Metasploitable2-Linux directory.

Linux Users: Most Linux distributions provide some form of a file browser not unlike the windows explorer, where students can *Copy* the Metasploitable2-Linux directory from one folder, and *Paste* it to another. However, if this is not the case, you may wish to use the terminal and the `cp -r` command instead.

The `cp`, or "copy" command is use to copy files from one location to another. The recursive "-r" option is required for copying folders and all their contents – it says "copy this entire folder and all of its content to the location I specify."

```
cp -r /home/[username]/Downloads/Metasploitable2-Linux /home/[username]/vmware
```

Replace the `[username]` portion of the command above with your username. For example, the username on my system is "ayy":

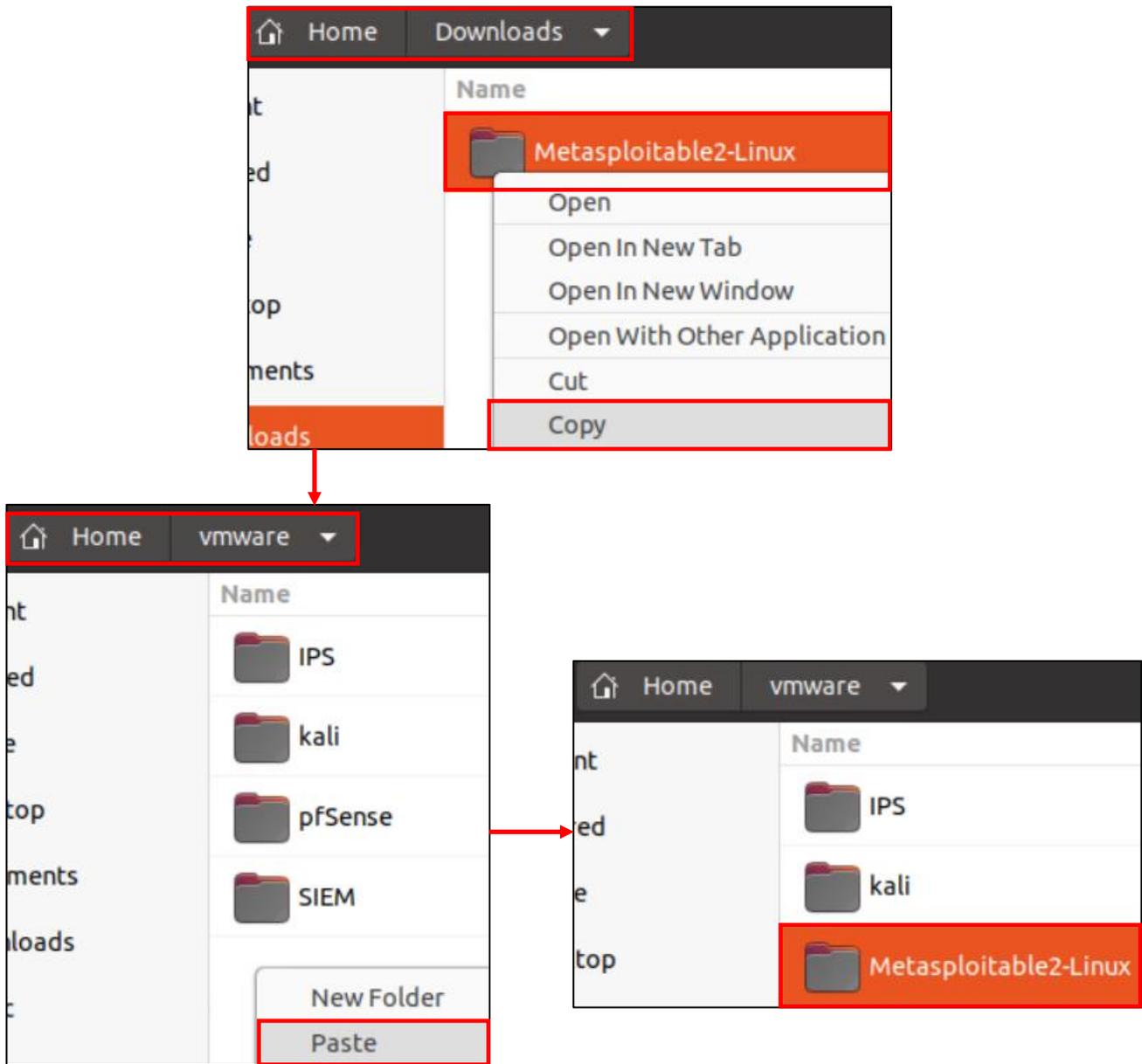
```
cp -r /home/ayy/Downloads/Metasploitable2-Linux /home/ayy/vmware
```

This command will copy the entire directory `Metasploitable2-Linux` from `/home/ayy/Downloads` to `/home/ayy/vmware`. Students may need to adjust the command according to where they stored the `Metasploitable2-Linux` folder, or the *Default location for virtual machines* they configured in the *Preferences* menu. Students can confirm the successful transfer of the `Metasploitable2-Linux` directory to the `vmware` directory by running the command:

```
ls -al /home/[username]/vmware
```

Once again, replace `[username]` with the username students use to log in to the host (e.g. "ayy"):

```
ls -al /home/ayy/vmware
```



12-68: Like Windows, Most Linux distros have some form of a file browser that allows for copying and pasting folders to move them around...

```

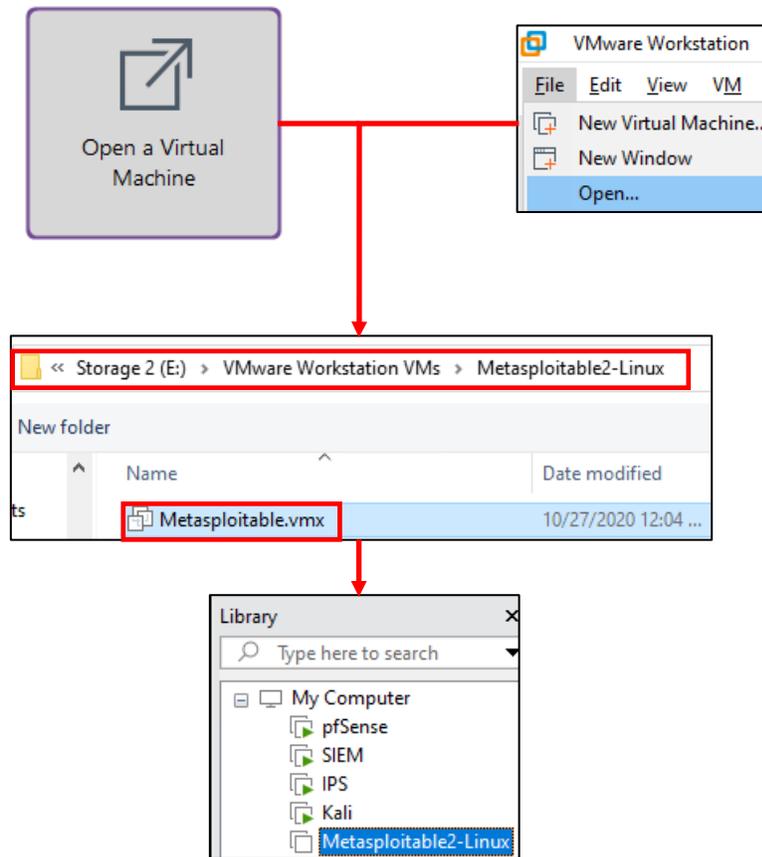
ayy@ayy:~/Downloads$ cp -r /home/ayy/Downloads/Metasploitable2-Linux/
/home/ayy/vmware/
ayy@ayy:~/Downloads$ ls -al /home/ayy/vmware
total 28
drwxrwxr-x  7 ayy ayy 4096 Oct 27 16:06 .
drwxr-xr-x 19 ayy ayy 4096 Oct 27 15:41 ..
drwxrwxr-x  2 ayy ayy 4096 Oct 27 15:54 IPS
drwxrwxr-x  2 ayy ayy 4096 Oct 27 15:54 kali
drwx----- 2 ayy ayy 4096 Oct 27 16:06 Metasploitable2-Linux
drwxrwxr-x  3 ayy ayy 4096 Oct 27 15:55 pfSense
drwxrwxr-x  2 ayy ayy 4096 Oct 27 15:54 SIEM

```

12-69: ...Alternatively, practically every Linux distro has the cp command for copying files and folders.

With the Metasploitable2-Linux directory relocated to the *Default location for virtual machines*, open VMware Workstation, and on the *Home* tab, click the option labeled *Open a Virtual Machine* (Optionally, students may use the navigation menu and select *File > Open* instead). Workstation will open the file explorer application for your host operating system. By default, it should open the file browser in the Default virtual machine folder students configured (e.g. C:\Users\[username]\Documents\Virtual Machines or in my case, E:\VMware Workstation VMs).

If this is not the case, navigate to the Default virtual machine folder manually, select the Metasploitable2-Linux directory, and double click on the Metasploitable.vmx file to select it. This will close the file browser, and a new virtual machine entry labeled Metasploitable2-Linux should appear in the *Library* pane.



12-70: Use either the Navigation menu or the huge *Open a Virtual Machine* button, locate the *Metasploitable.vmx* file located in the *Metasploitable2-Linux* directory that you moved to the default virtual machine directory. This will cause VMware Workstation to register a new VM named *Metasploitable2-Linux* in the *Library* pane.

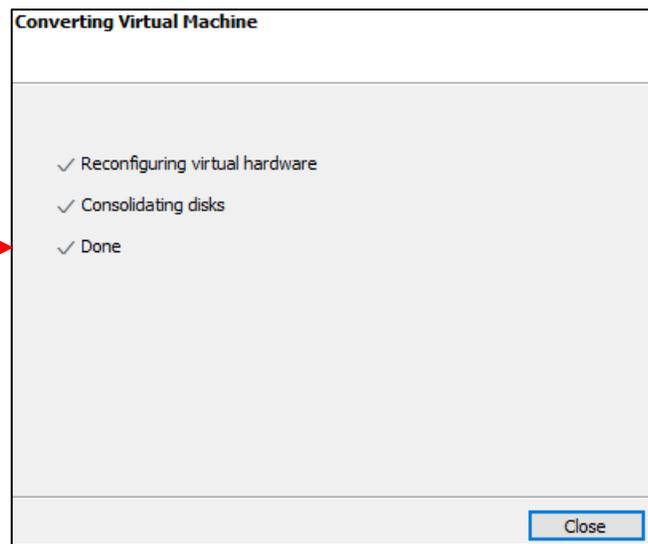
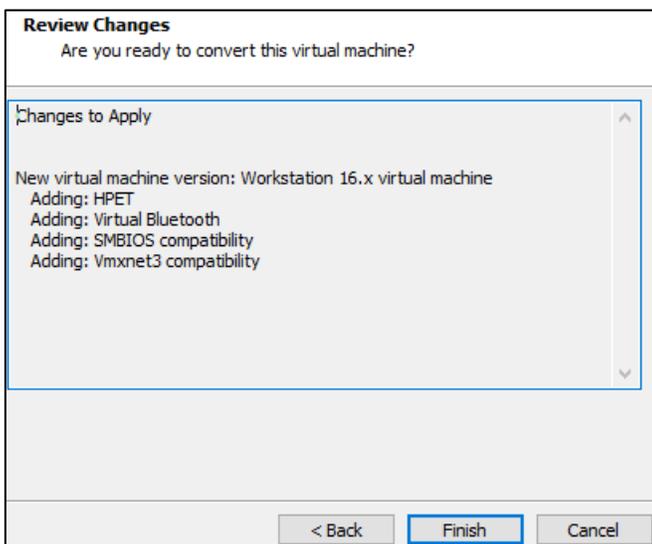
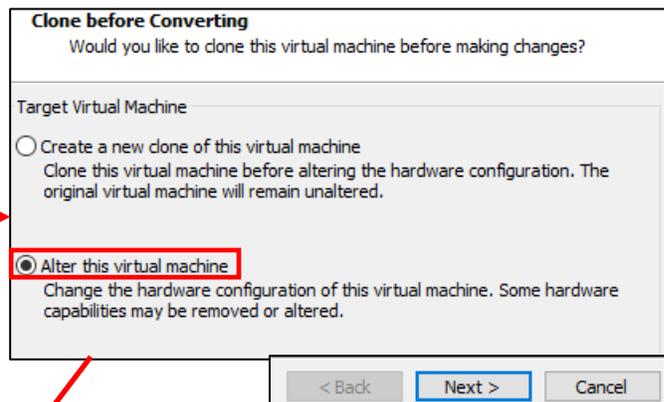
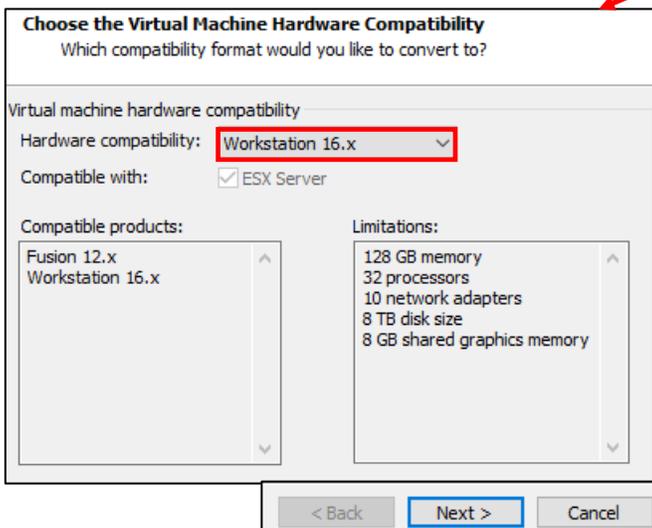
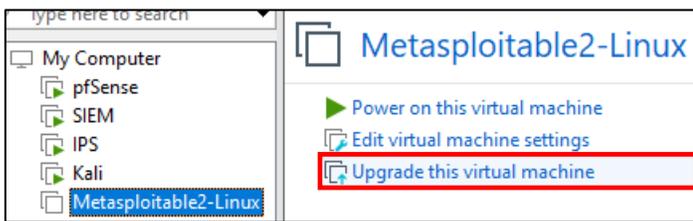
12.6.4.1 Upgrading the Metasploitable 2 VM

With the Metasploitable 2 VM registered to VMware Workstation, the next step is upgrading its virtual hardware. To do that, click on Metasploitable2-Linux in the *Library* pane, then click on the text *Upgrade this virtual machine*. This opens a new window titled *Change Hardware Compatibility Wizard*. Click the *Next* button to proceed.

The next screen, labeled *Choose the Virtual Machine Hardware Compatibility* appears. In the drop-down labeled *Hardware compatibility*, choose *Workstation 16.x* (or the current version of VMware Workstation, if you are using a later release), then click *Next*.

A new screen appears labeled *Clone before Converting*. Workstation gives users the choice to create a new clone of the current virtual machine in the event that upgrading the virtual hardware causes some incompatibilities. Fortunately, students should have the original zip file used to create this virtual machine, so there should be no need to create a clone and waste disk space. Click the radio button labeled *Alter this virtual machine*, and click *Next*.

The final screen, labeled *Review Changes* asks users to confirm their changes. Click the *Finish* button to proceed, then click the *Close* button on the *Converting Virtual Machine* screen to exit the wizard and upgrade the Metasploitable 2 VM.

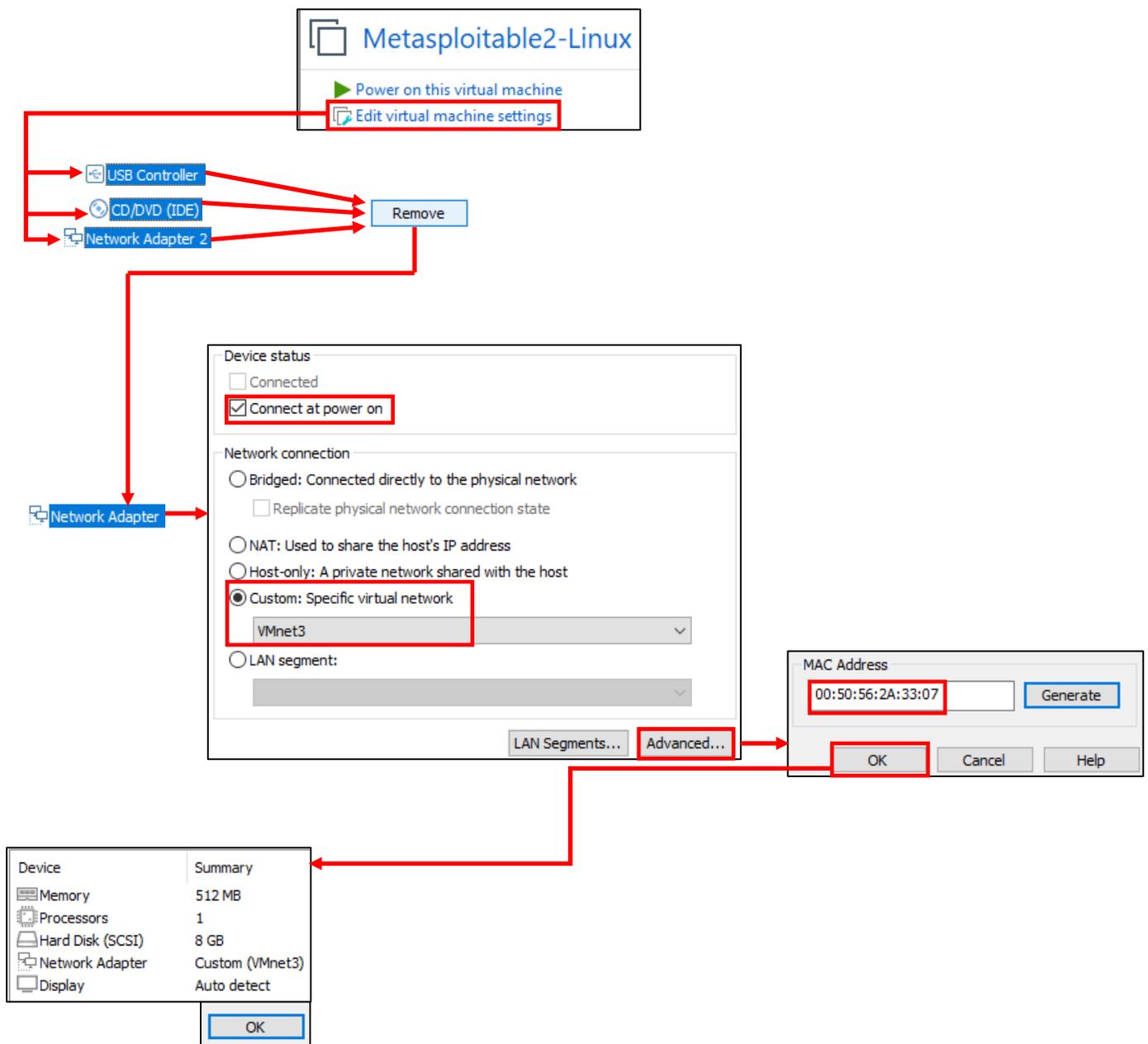


12-71: Highlight the Metasploitable2-Linux entry and click *Upgrade this virtual machine*. The Change Hardware Compatibility Wizard is pretty straight forward. Make sure to choose Workstation 16.x (or the current version of VMware Workstation) for the *Hardware compatibility* drop-down on The *Choose the Virtual Machine Hardware Compatibility* screen, and the *Alter this virtual machine* radio button on the *Clone before Converting* screen.

12.6.4.2 Edit Metasploitable 2 Virtual Machine Settings

Highlight the Metasploitable2-Linux entry on the *Library* pane again. This time, select *Edit virtual machine settings*. The nice thing about Metasploitable 2 is that the operating system is pre-installed, so students will be able to perform all of the *Virtual Machine Settings* edits in one go. Perform the following configuration changes:

- Remove the *CD/DVD (IDE)*, *USB Controller*, and *Network Adapter 2* virtual hardware devices
- Highlight the *Network Adapter* device. Under the *Network connection* portion, select *Custom: Specific virtual network*, and select *VMnet3 (/dev/vmnet3)* from the drop-down menu. Ensure the *Connect at power on* checkbox under *Device status* is checked.
- With the *Network Adapter Device* still highlighted, click the *Advanced* button to open the *Advanced Network Adapter Settings* window. Generate and document the newly generated MAC address of the Metasploitable 2 VM and the network segment it is attached to. Click *OK* to close the *Advanced Network Adapter Settings* window, then click *OK* again to close the *Virtual Machine Settings* menu.
- Open a session to the pfSense WebConfigurator, navigate to *Services > DHCP Server > OPT1* and create a static DHCP mapping for the newly recorded MAC address of the metasploitable 2 VM, assigning it the IP address 172.16.2.3. **It is extremely important that the Metasploitable 2 VM always gets the IP address 172.16.2.3.** Apply your changes.



Services / DHCP Server / OPT1

DHCP Static Mappings for this Interface				
Static ARP	MAC address	IP address	Hostname	Description
	00:50:56:21:9e:c3	172.16.2.2	kali	static DHCP mapping for kali VM
	00:50:56:2a:33:07	172.16.2.3	metasploitable2	static DHCP mapping for Metasploitable 2 VM

12-72: Open the *Virtual Machine Settings* menu for the Metasploitable2-Linux VM. Remove the CD/DVD (IDE), USB Controller and Network Adapter 2 Devices. Highlight the Network Adapter, and ensure the *Connect at power on* Device status checkbox is checked. Under *Network connection*, select the *Custom: specific virtual network* radio button, then select *VMnet3* (/dev/vmnet3) from the drop-down menu. Click *Advanced*, Generate a MAC address, document the VM it belongs to, and the network it is attached to. Log into the pfSense WebConfigurator and create a new static DHCP mapping for the Metasploitable 2 VM on the *OPT1* interface. ***Assign the Metasploitable 2 VM the IP address 172.16.2.3. This is extremely important.***

12.6.4.3 Metasploitable 2 Test Run

Power on the Metasploitable 2 VM, and connect to its virtual console. A whole bunch of text will scroll by as the VM goes through the boot process. After some time has passed, students should be greeted with a login prompt. The default credentials for metasploitable 2 are the username and password combination of `msfadmin/msfadmin`. Upon logging in, run the command `ifconfig -a`, and confirm that the interface `eth0` appears. **Record the contents of the field labeled `HWaddr`, the MAC address of `eth0`.** Confirm it matches the MAC address of the static DHCP mapping students just created for the Metasploitable 2 VM. When finished, type `exit` to log out of the virtual machine.

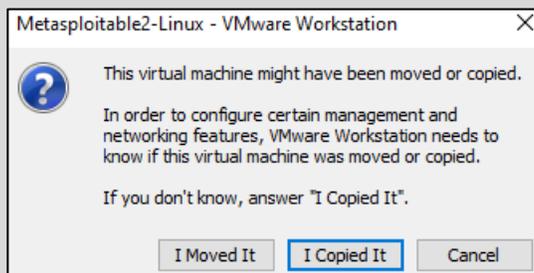
The screenshot shows a terminal window for a Metasploitable 2 VM. The user has logged in as `msfadmin` and run the command `ifconfig -a`. The output shows the configuration for the `eth0` interface, with the `HWaddr` field highlighted in red. A red arrow points from this field to a table below the console output.

Services	DHCP Server	OPT1
00:50:56:2a:33:07	172.16.2.3	metasploitable2

12-73: Power on the Metasploitable VM, and connect to its virtual console. Login with the username/password of `msfadmin/msfadmin`. Run the command `ifconfig -a`, record the `HWaddr` field for the `eth0` interface. Confirm it matches the MAC address of the static DHCP mapping students just created for the Metasploitable 2 VM then type `exit` to log out.

Who Touched my VM?

When you first power on the Metasploitable 2 VM, you'll be greeted by a pop-up that demands to know whether you moved or copied the VM before it will start up. Click the button labeled *I Copied It* to proceed.



12-74: Who touched Sasha? WHO TOUCHED MY VM?

Why aren't we doing connectivity checks?

Some of you may be wondering why we aren't doing connection checks or any of the stuff we did we for the SIEM, IPS, or Kali VMs, like checking that the static DHCP allocation is working, or attempting to connect outbound. Well, that's because right now, the metasploitable 2 VM doesn't have an IP address at all. Don't worry, its intentional, and you'll be fixing this later. The reason metasploitable 2 doesn't have an IP address is that its connected to the *VMnet3* (IPS2) network. While technically the *VMnet3* (IPS2) shares the same network subnet as *VMnet2* (IPS1), and logically it's all a part of the *OPT1* network, IPS2 is its own physical network segment, and entirely separate from the IPS1 network. **Without something to bridge connect the IPS1 and IPS2 networks together, the IPS2 network is entirely isolated.**

Remember the network diagram back in [chapter 6](#) (p. 58)? The *IPS2* network relies on the IPS virtual machine being fully configured and running either Snort or Suricata in AFPACKET bridging mode. No network bridge, no network connectivity. That means no IP address from the DHCP server, either. You can see this for yourself from the output of `ifconfig -a`. You'll be fixing this later when you install either Snort or Suricata to the IPS virtual machine in [chapter 17](#).

```
msfadmin@metasploitable:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:50:56:2a:33:07
          inet6 addr: fe80::250:56ff:fe2a:3307/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2520 (2.4 KB)
          Interrupt:17 Base address:0x2000
```

12-75: `eth0` never got assigned an IPv4 address from DHCP because there is no physical connectivity between the IPS1 and IPS2 networks (*VMnet2* and *VMnet3*, respectively). We'll be solving this problem later when students install Snort or Suricata to the IPS virtual machine.

12.7 Snapshots

The next (and final) task for students to perform will be creating baseline virtual machine snapshots for the entire lab environment. Snapshots (sometimes referred to as checkpoints by other hypervisors) instruct the virtual machine's hypervisor to gather information about the VM's current state, and save it. Later on, if there is a problem with the virtual machine such as a malware infection, or a configuration problem that cannot be diagnosed, users can choose to restore the virtual machine to its state in the past, when the snapshot was initially created

Snapshots can be created with virtual machines powered off, or while they are running, making them extremely versatile. VMware Workstation virtual machines can also have more than one checkpoint, with the only limit being disk space required to hold them. It's extremely important to note that **virtual machine snapshots are not a substitute for backups**. If students plan on running virtual machines with important data that they cannot afford to lose, snapshots are not a substitute for backing up important files and data.

In this section, students will walk through the process of creating a virtual machine snapshot for the pfSense VM. Afterwards, it will be left as an exercise to the students to repeat the process for the SIEM, IPS, kali, and Metasploitable 2 virtual machines. Once finished, students will be ready to move on with the configuration of their lab environment.

12.7.1 How to Create a Snapshot

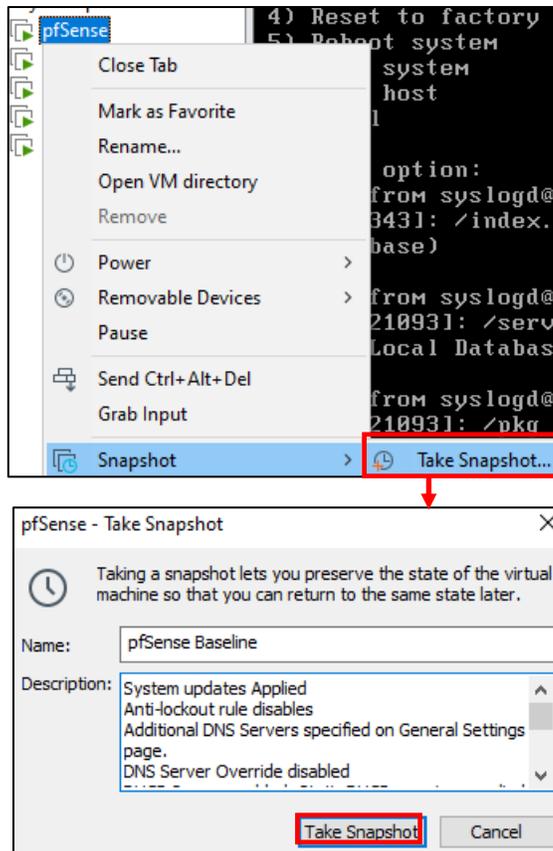
In the *Library* pane of the VMware Workstation interface, right-click on the pfSense listing. Select the option labeled *Snapshot*, then *Take snapshot* from the sub-menu that appears. A window labeled *pfSense - Take Snapshot* will appear. In the *Name* field, enter a name for this snapshot that provides a brief description about the state of the virtual machine. The *Description* field can be used to provide more detailed information about the state of the VM. For example, I recommend entering the following *Name* and *Description* for the pfSense VM's first snapshot:

Name: pfSense Baseline

Description:

System updates Applied
Anti-lockout rule disabled
Additional DNS Servers specified on General Settings page.
DNS Server Override disabled
DHCP Server enabled, Static DHCP mappings applied
DNS Resolver service enabled for LAN and OPT1
Squid proxy service installed and enabled for LAN and OPT1
NTP enabled for LAN and OPT1
Firewall policy applied for WAN, LAN and OPT1

Once finished, click the *Take snapshot* button for Workstation to begin creating the snapshot.

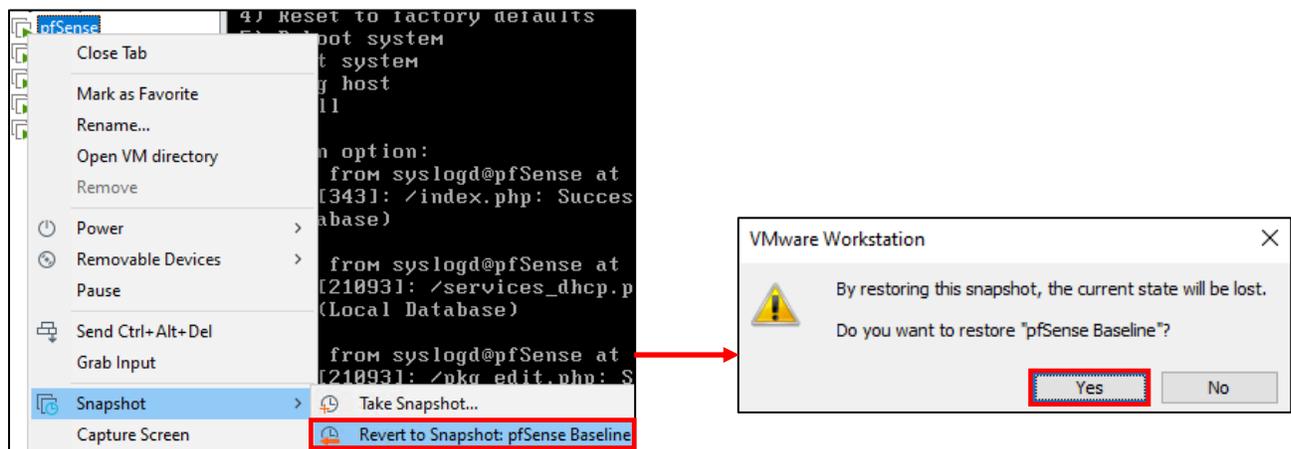


12-76: Right-click on the pfSense entry on the *Library* pane. Select *Snapshot*, followed by *Take snapshot*. In the window that appears enter a name and description for your snapshot, then click the *Take snapshot* button. Use descriptive names, and add detailed descriptions.

12.7.2 Restoring a Snapshot

In order to restore a virtual machine snapshot, right-click the target VM in from the list on the *Library* pane, select *Snapshot*, followed by *Revert to Snapshot: [snapshot name]* in the sub-menu that appears. A warning window appears stating that the current state of the VM (that is, any changes made to the VM since the snapshot was taken) will be lost. The pop-up asks if users want to continue restoring their snapshot. Click the *Yes* button to continue.

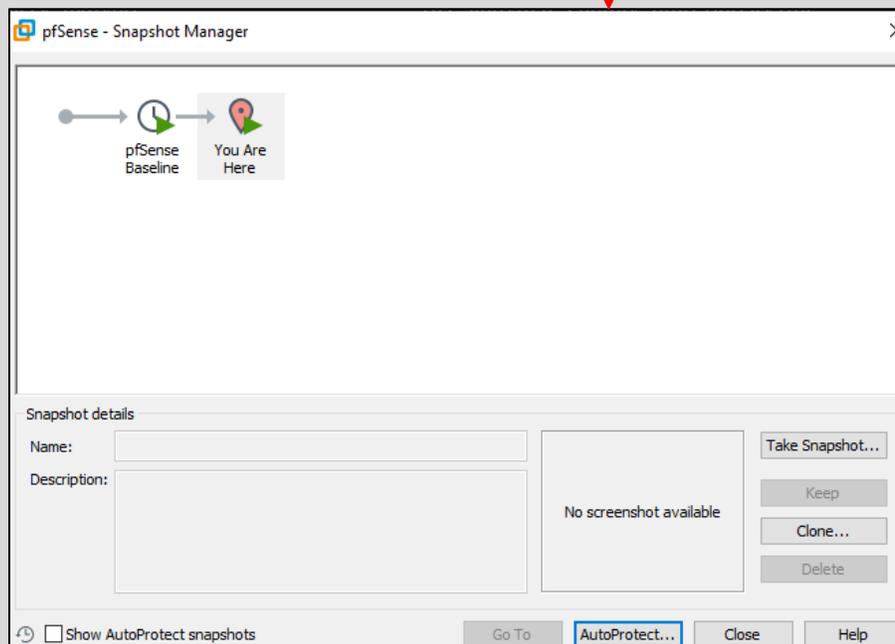
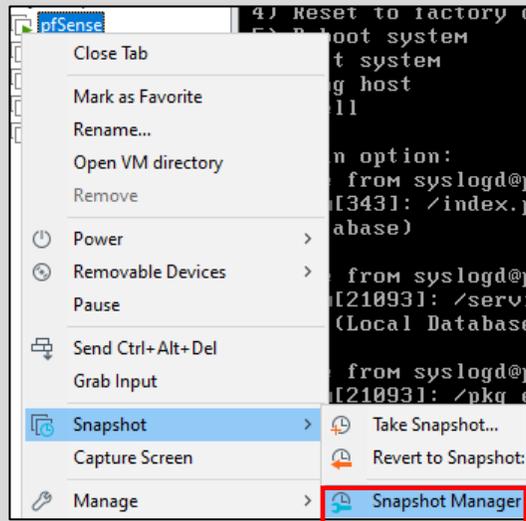
The Revert to snapshot option only allows students to restore to the last snapshot taken on the target virtual machine. Currently this isn't a problem, because the pfSense VM only has a single snapshot. Students are recommended to read the sidebar conversation below *What if I have more than One Snapshot* to learn how to better manage multiple snapshots on a single virtual machine.



12-77: Restoring snapshots is even easier than making them. Right-click on the target VM, select the *Snapshot* option, followed by *Revert to Snapshot: [snapshot name]*. On the window that pops up, click *Yes*. After a few moments, the hypervisor will restore the virtual machine's state. Have more than one snapshot? Want to restore a VM to a specific state? Want to de-clutter and remove old snapshots? Read the sidebar discussion below.

What if I have more than One Snapshot?

Clicking the *Revert to snapshot* option only allows students to restore the most recent snapshot they created. That's fine if all you have is a single snapshot for your VM, but what if you have more? Right-click on the target virtual machine in the Library pane, select *Snapshot*, followed by *Manage snapshots*. This will cause a new window to appear titled *[VM name] – Snapshot Manager*. From here, you can create additional snapshots, select specific snapshots to revert to, delete snapshots you no longer have a need for, etc.



12-78: Access the snapshot manager for the target virtual machine, by right clicking on it on the Library pane, selecting *Snapshot*, then *Snapshot Manager*. The snapshot manager allows you to create as many snapshots as you please (so long as there is disk space available), restore to any previously taken snapshot, or delete snapshots that

12.7.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines

Now that students understand how to create snapshots, it is highly recommended that they create baseline snapshots for the remaining virtual machines in their lab environment – the SIEM, IPS, kali, and metasploitable 2 virtual machines. In the chapters to come, there will be a lot of complicated configuration tasks that students will need to perform in order to enable different functionality for their environment. Having a baseline snapshot to fall back to in case there are problems completing a task is handy for troubleshooting purposes.

SIEM - Take Snapshot	
	Taking a snapshot lets you preserve the machine so that you can return to the sa
Name:	SIEM Baseline
Description:	OS installation complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied Splunk not yet installed

IPS - Take Snapshot	
	Taking a snapshot lets you preserve the state of the virt machine so that you can return to the same state later.
Name:	IPS Baseline
Description:	OS installation complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied Splunk Forwarder and IDS/IPS software not installed

Kali - Take Snapshot	
	Taking a snapshot lets you preserve the machine so that you can return to the sa
Name:	kali baseline
Description:	Initial OS install complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied

Metasploitable2-Linux - Take Snapshot	
	Taking a snapshot lets you preserve the state of machine so that you can return to the same state
Name:	Metasploitable 2 Baseliem
Description:	Successfully imported and updated Confirmed VM powers on and successful login Static DHCP mapping created, not yet tested No network connectivity (yet)

12-79: Now that students know how to create snapshots, apply that knowledge and make baseline snapshots for the other lab virtual machines. Having a baseline to fall back to in case something fails in the later chapters of this book is very important and will save students from a lot of headaches.

12.8 Chapter Review

Students should have all 5 virtual machines created for the baseline lab environment, as well as baseline checkpoints for all 5 virtual machines. It was a long journey to get to this point, but it's far from over. Here is a checklist of tasks to complete:

- Complete chapter 15, *Routing and Remote Access for Hosted Hypervisors*, starting on p. 727. In this chapter, students will learn how to enable SSH access to their lab virtual machines from Windows, Linux or MacOS. This functionality is vital for finishing the IPS and Splunk setup guides more easily than through the VM console alone.
- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in chapter 17, *Network Intrusion Detection*, starting on p. 980.
- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in chapter 18, *Setting up Splunk*, starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out chapter 19, *End of the Beginning*, starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out chapter 20, *Extra Credit*, starting on p. 1055.

Chapter 13 Patch Notes

- A lot has changed with ESXi over the past few years. Most notably, the HTML5 Web interface is now the default. That means that the specific Windows ESXi client is no longer covered.
- Since the HTML5 interface is now standard, that means that VMware Flings (e.g. the experimental packages repo for ESXi) will not be covered, either. Curious students can look into VMware Flings on their own, if they want to mess with the experimental stuff, but overall, this is a good thing as its once less thing to mess with the stability of the hypervisor, and one less complex task to complete
- Dedicated more time to talking about hardware requirements for ESXi, including a lot of discussions centered around hardware compatibility and how sometimes, just because the hardware is compatible, not all of the hardware features are supported.
- Dedicated entire sections how to install UNetbootin on Windows, Linux, and MacOS, as well as how to use it, once installed, to create a bootable ESXi USB drive
- Acknowledge that ESXi is extremely picky, and tried to answer why students would want to persist in setting up ESXi even in spite of how notoriously finicky it is. Short answer is that its enterprise software, and getting exposed to it now prepares you to fight it later. Also discussed the benefits in helping to build technical support and troubleshooting prowess
- Provided alternative bare-metal hypervisor recommendations that are not so picky hardware-wise if students want an alternative to ESXi
- Because I'm supremely lazy, and I know many of you are too, took some time to show students how to disable the web interface timeout that, by default kicks users off of the ESXi web interface after 15 minutes of inactivity. Yay checkbox compliance
- Talked a bit more about why its recommended the ESXi server has at least two physical network interfaces – one for carrying VM traffic to/from physical networks, and the other dedicated to handling management/administrative traffic (e.g. traffic to the web interface)... and how to get by with just a single network interface if absolutely necessary
- Dedicated a section on how to create a datastore on the ESXi web interface, as well as discussing best practices for datastores, and troubleshooting storage problems
- Warned students that configuring virtual disks with the *Thick Provision, eagerly zeroed* configuration does take longer, but offers better performance compared to other virtual disk options
- Informed users that ESXi is lazy about MAC address assignment, and waits until after the first time the VM is booted before assigning MAC addresses to assigned interfaces

- Described a problem where attempting to remove the *SATA Controller 0* virtual hardware, even after removing *CD/DVD Drive 1* (the only device attached to it) will throw an error saying that the CD/DVD drive is still present when students attempt to remove it, and how to work-around it
- Apparently `pfctl -d` no longer works to disable the pfSense firewall, and allow users to access the pfSense firewall and reconfigure it to allow access from the WAN interface. Instructed users to utilize the `pfSsh.php enableallowallwan` method instead to gain initial access. Warned that this leaves the pfSense wide open and to get on the bloody thing and change the admin password as soon as possible, after issuing this command
- ESXi refuses to recognize Kali Linux as a flavor of Debian, and insists on telling users that the guest OS setting is wrong. I tell the students "no, it's the hypervisor that is wrong" but if they insist on changing the setting for troubleshooting purposes, how it can be modified to stop the hypervisor from complaining
- Provided students two different methods for uploading the Metasploitable 2 VM to their ESXi server: The first method is the traditional "Let the official VMware vCenter Converter app do it for you", while the second method is considerably more difficult. I'll refer to it as the "manual method".
- Discovered an issue with the "manual method" where by attempting to perform snapshot operations against Metasploitable 2 while the VM is running will fail. The VM doesn't crash, but it just refuses to allow users to manage snapshots for the VM (creation, restoring from, deleting, etc.) unless the VM is powered off.

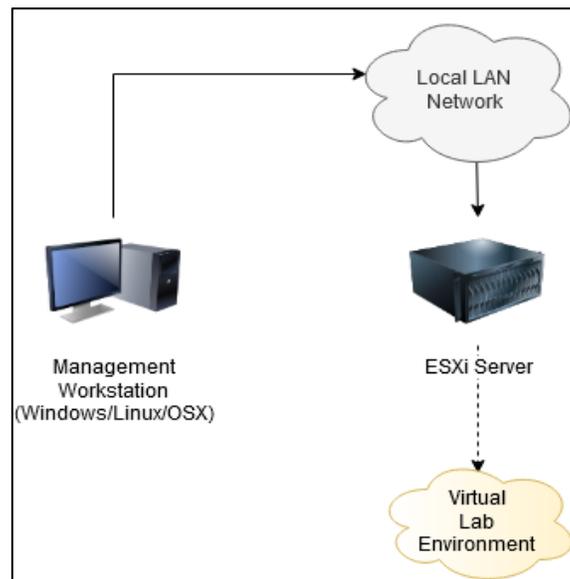
Chapter 13: ESXi

VMware ESXi is probably one of the most well-known and popular bare-metal hypervisors out there today. It's used in enterprise environments and home lab networks alike all over the world. Over the years, the hypervisor has been given many names (ESX, ESXi, VMware vSphere Hypervisor, etc.), but for the sake of simplicity, it will be referred to as ESXi. **Please note that the current (free) release of ESXi when this guide was written was version 7.0.**

13.1 Prerequisites

Way back in Chapter 3, [section 3.2.2](#) (pp. 42-43), students learned about what makes bare-metal hypervisors unique. In a nutshell, a bare-metal hypervisor runs directly on top of hardware with a minimal operating system. This operating system has just enough functionality to manage the hardware and virtual machines, provide some sort of a command-line interface to assist with troubleshooting problems on the system, and some sort of a network-based interface for administration of the hypervisor and its virtual machines from another workstation. This means that if students are interested in running ESXi for their lab environment, they will need to have at least two physical hosts:

- One host that meets the compatibility and resource requirements of ESXi, with the necessary resources to run and host the virtual lab environment.
- One host running some form of a desktop operating system (BSD, MacOS, Windows, Linux, etc.) with an HTML5 compatible web browser and network connectivity required to manage the ESXi server.



13-1: Most bare-metal hypervisors require their own dedicated, physical hardware to run, as well as a management workstation to manage the hypervisor itself. ESXi has a web interface for managing the hypervisor. So long as your desktop operating system has a web browser with HTML5 support, it doesn't matter what desktop operating system you use.

13.1.1 Installation Requirements

VMware provides a knowledgebase with all sorts useful documentation Check out:

<https://docs.vmware.com>

In the search bar, search for the term "esxi hardware requirements" and the first search result will take you here:

<https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.esxi.install.doc/GUID-DEB8086A-306B-4239-BF76-E354679202FC.html>

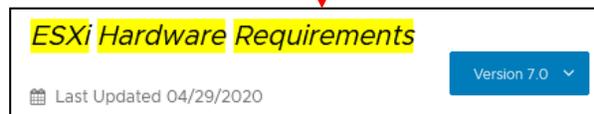
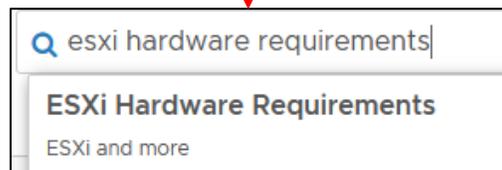
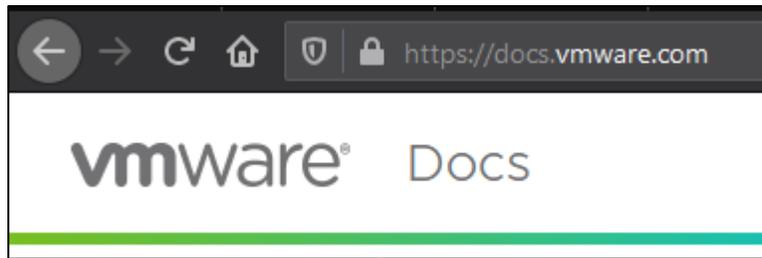
None of the hardware or resource requirements listed should come as a surprise, since students learned about hardware recommendations in chapter 5 (e.g., multi-core CPUs, 64-bit support, NX/XD bit support, VT-x/AMD-V) and recommended resource allocations in chapter 6 (e.g. 16GB+ RAM, 500GB+ disk space). If students followed the guidance in those chapters, then their system should more than beat minimum hardware requirements necessary to run ESXi. However, there are two very important details that need to be covered:

1. The very first bullet under the *Hardware and System Resources* heading mentions supported server platforms. This means picking hardware that is compatible with ESXi. ESXi is extremely picky about the hardware it runs on, but fortunately there are many people who run the hypervisor on hardware that isn't "officially" supported. We'll be talking more about this in section 13.1.2.

2. Pay attention to the bullet point:

ESXi 7.0 requires a boot disk of at least 8 GB for USB or SD devices, and 32 GB for other device types such as HDD, SSD, or NVMe. A boot device must not be shared between ESXi hosts.

This means that the ESXi hypervisor and operating system is lightweight enough to where it can be installed on a USB drive or SD card that has at least 8GB of storage space or more. If you don't have any USB drives around or would rather install it to a solid-state/NVMe or standard hard disk, they recommend at least 32GB of space instead. I highly recommend taking advantage of this feature and installing the hypervisor to either a USB drive or SD card. The primary advantage is that this maximizes the amount of disk I/O available to the virtual machines and is overall better for performance. In this chapter students will be installing ESXi to either an USB drive, or an SD card. **Students should ensure that they have access to at least two USB drives 8GB in size or higher, or 1 USB drive, and 1 SD card.**



To install or upgrade ESXi, your **hardware** and system resources must meet the following requirements:

- Supported server platform. For a list of supported platforms, see the *VMware Compatibility Guide* at <http://www.vmware.com/resources/compatibility>.
- **ESXi** 7.0 requires a host with at least two CPU cores.
- **ESXi** 7.0 supports a broad range of multi-core of 64-bit x86 processors. For a complete list of supported processors, see the VMware compatibility guide at <http://www.vmware.com/resources/compatibility>.
- **ESXi** 7.0 requires the NX/XD bit to be enabled for the CPU in the BIOS.
- **ESXi** 7.0 requires a minimum of 4 GB of physical RAM. Provide at least 8 GB of RAM to run virtual machines in typical production environments.
- To support 64-bit virtual machines, support for **hardware** virtualization (Intel VT-x or AMD RVI) must be enabled on x64 CPUs.
- One or more Gigabit or faster Ethernet controllers. For a list of supported network adapter models, see the *VMware Compatibility Guide* at <http://www.vmware.com/resources/compatibility>.
- **ESXi** 7.0 requires a boot disk of at least 8 GB for USB or SD devices, and 32 GB for other device types such as HDD, SSD, or NVMe. A boot device must not be shared between **ESXi** hosts.
- SCSI disk or a local, non-network, RAID LUN with unpartitioned space for the virtual machines.
- For Serial ATA (SATA), a disk connected through supported SAS controllers or supported on-board SATA controllers. SATA disks are considered remote, not local. These disks are not used as a scratch partition by default because they are seen as remote.

13-2: docs.vmware.com has tons of info on VMware products, including hardware requirements for ESXi. Recommended system features and resource allocations were covered in Chapters 5 and 6. Those resource recommendations should be more than enough to satisfy the minimum hardware requirements for ESXi.

Boot from External Media

In recent years, many hardware manufacturers have gotten wise to the fact that most organizations run ESXi or (other hypervisors) and make full use of its ability to boot from, and run on an SD card, or USB drive. As a result, some server manufacturers provide a dedicated USB port inside of the server chassis, or may even provide dedicated SD card slots. If you managed to acquire server hardware somehow, keep an eye out for SD card slots or USB ports inside the chassis, or located on the front access panel of the server.



13-3: This is an image taken by Andreas Lesslhuber, on his blog, running-system.com. This image demonstrates how server manufacturers sometimes provide SD, micro-SD, and/or USB ports inside of the server chassis for installing bare-metal hypervisors, or other lightweight operating systems. Some vendors even provide functionality that allows administrators to configure SD cards into a RAID array for further redundancy. See that black object with the Hewlett Packard logo plugged into the bottom USB port? It's a module provided by HP that allows administrators to plug in 2 SD cards. The module will automatically configure them into a RAID 1 array.

Out of Band/IPMI

Many server manufacturers provide support for some form of Out of Band (OOB) management. Each vendor has a different name for this technology – IPMI, iLO, iDRAC, CIMC, ALOM, etc. In a nutshell, these are dedicated network interface ports on some server hardware that allows system administrators, through the user of a web browser, or installed software to control the server over the network as though their keyboard, video and mouse were directly connected to the server. Some of these Out of Band (OOB) management platforms even allow admins to mount an ISO directly to the server over a network connection, and boot from it.

As awesome as these features are, not every student who wishes to complete this chapter will have access to this type of hardware and software. If you have access to OOB management software, and you know how to use it to mount an installation ISO, feel free to use it. However, **I will be operating under the assumption that all students will be installing ESXi from a USB drive, and installing the operating system to either an SD card, or another USB flash drive. This means that students will need either two USB flash drives, or a USB flash drive, and an SD card (if your system has an SD card slot) in order to install ESXi.** One USB flash drive will be used to hold the VMware installation files, while the other flash drive (or SD card) will be the disk students install ESXi to.



Refresh | Host Power | Launch KVM | Ping | Reboot | Locator LED |

Cisco Integrated Management Controller (Cisco IMC) Information

Hostname:	RagnorokCIMC
IP Address:	10.0.0.4
MAC Address:	00:BE:75:C8:05:38
Firmware Version:	3.0(3c)
Current Time (UTC):	Wed Sep 2 09:07:08 2020
Local Time:	Wed Sep 2 09:07:08 2020 UTC +0000
Timezone:	UTC Select Timezone

13-4: The Cisco UCS server in my basement has this powerful management controller software that runs in the web browser. That little Launch KVM option up there is the best thing for server administration ever. However, not every student will be installing ESXi on actual server hardware, nor can it be guaranteed that they will have access to software and features like this. Instructions for installing ESXi in this chapter will not rely on these features.

13.1.2 Hardware Compatibility

When it comes to ESXi's hardware compatibility requirements, its notorious for being *very* finicky, and flat-out refusing to work with a lot of hardware. It's not uncommon for someone to try and take an old workstation or PC gaming system, attempt to convert it into an ESXi server, and fail due to the hardware being unrecognized.

For those who are serious about running ESXi as their preferred hypervisor, and forcing their hardware into submission, consider making use of the *VMware Compatibility Guide*:

If you're going to try to build your own server, or retrofit an old workstation/gaming PC into a server, the most important things to know are whether or not ESXi supports are the disk controller (usually the SATA controller for most desktop systems), and the ethernet controller (aka the network interfaces). VMware provides a tool called the *VMware Compatibility Guide*. Specifically, students can search for I/O Devices that are compatible with ESXi. The current link to this resource is:

<https://www.vmware.com/resources/compatibility/search.php?deviceCategory=io>

Alternatively, if using a pre-built server (e.g., Dell, HP, Cisco, etc.), VMware also provides a Compatibility Guide for server makes and models as well. The current link to this resource is:

<https://www.vmware.com/resources/compatibility/search.php>

VMware Compatibility Guide

The screenshot displays the VMware Compatibility Guide search interface. At the top, there is a search bar with the text "Search Compatibility Guide: ? (e.g. compatibility or esx or 3.0)" and a dropdown menu set to "All Listings". A "Search" button is located to the right. Below the search bar, the interface shows "What are you looking for: IO Devices" and "Compatibility Guides" with a "Help" link and "Current Results: 0". The main area contains several filter categories, each with a list of options and a dropdown arrow:

- Product Release Version:** All, ESXi 7.0, ESXi 6.7 U3, ESXi 6.7 U2, ESXi 6.7 U1, ESXi 6.7
- Brand Name:** All, Adaptec, Adaptec by PMC, Advantech Corporation, Allied Telesis
- Keyword:** L8200A
- I/O Device Type:** All, Block, FC, FCoE CNAs, Hardware Acceleration, iSCSI, Memory Channel Attached Storage (MCA), Network, NVMe, PATA, SAS
- Driver Types:** All, Partner Async, VMware Inbox
- Features:** All, 4K, 512e, 512e, DIF/DIX (Type 1), Enhanced data path, Enhanced data path - Poll mode, Firmware NetDump, GENEVE-Offload, GENEVE-RxFilter, IPv6
- VID:** All
- DID:** All
- SVID:** All
- Max SSID:** All
- Posted Date Range:** All

At the bottom, there are two buttons: "Update and View Results" and "Reset".

13-5: The VMware Compatibility Guide interface.

Picky Eater

As an exercise, I searched the internet for a recently released desktop gaming motherboard to try and determine whether or not it would be compatible with ESXi. ASUS is usually my preferred motherboard manufacturer when I'm going to build a gaming PC, so for this exercise, I decided to pick on the "TUF Gaming X570-Plus". Searching google for "TUF Gaming X570-PLUS" brings us to the manufacturer's support site, and motherboard documentation page:

https://www.asus.com/motherboards-components/motherboards/tuf-gaming/tuf-gaming-x570-plus/HelpDesk_Manual/

Looking at the user guide, there is a specifications summary page. The LAN adapter is a Realtek L8200A, and the SATA controller, listed under Storage is provided by the AMD X570 chipset. Searching for either device in the compatibility guide for IO devices returns absolutely nothing, which is par for the course.

Google (or your preferred search engine) is going to be your best friend here. Searching for "AMD X570 ESXi" returns results where people confirm that yes, the X570 chipset was recognized and supported properly in the latest version of ESXi. Unfortunately, searching for "Realtek L8200a ESXi" doesn't really provide any results that seem to indicate that ESXi has the driver by default.

However, further searching suggests that there are a couple of guides and websites out there that may be able to help you. Be aware that some of these blog posts may be dated, and may or may not solve your problems. But if you're determined on getting your hardware to work in ESXi, they may be right for you:

v-front has a blog post on how to see if perhaps there is a community-provided driver for your device here: <https://www.v-front.de/2014/12/how-to-make-your-unsupported-nic-work.html>

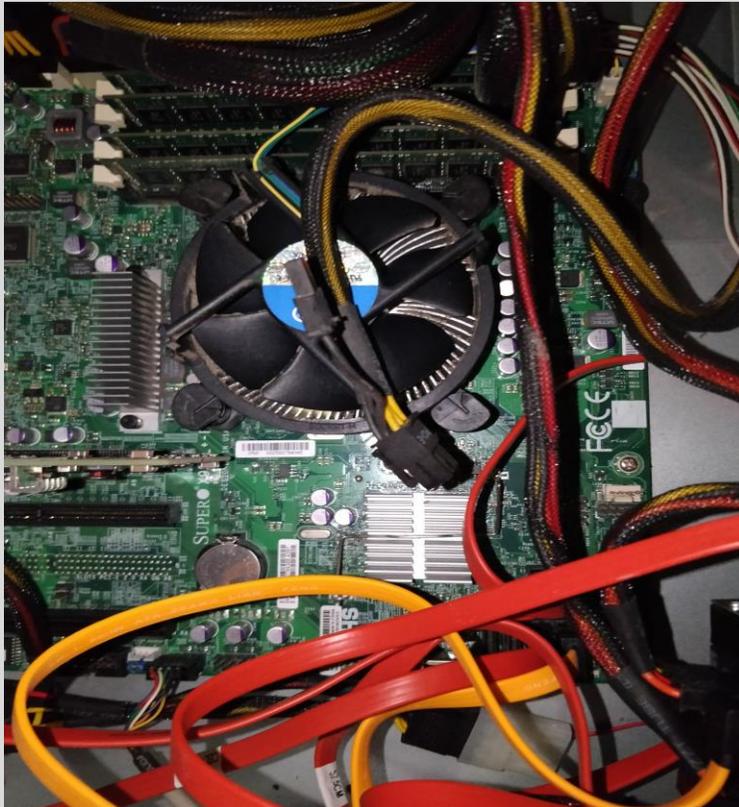
virtuallyGhetto is another resource that seems to be dedicating towards forcing hardware that doesn't want to work with ESXi into submission: <https://www.virtuallyghetto.com/>

Now, if mucking about with driver packages and making custom installation ISOs appears to be outside of your comfort zone, then then the safer alternative is to make use of expansion slots and install compatible hardware. As far as network cards go, ESXi works really well with Intel NICs, so if you're going to go this route, I recommended purchasing them. It looks like that motherboard I'm picking on has one PCI express 16 slot, and three PCI express X1 slots. I would recommend purchasing PCI express x1 network card with more than one network interface, that supports gigabit ethernet, or better. For example, Intel's I350-T4 adapter fits the slot, and has four gigabit ethernet ports. Alternatively, you could install multiple PCI express x1 network cards, like the Intel EXPI9301CTBLK.

Bear in mind, this can be an expensive proposition. That I350-T5 card, as of mine writing this in September of 2020 was going for ~130.00 USD. Meanwhile, the EXPI9301CTBLK goes for ~35.00 USD.

As a final note, and bearing from my personal experience, **just because ESXi recognizes a piece of hardware does not mean that all of its functionality is supported.** Let me elaborate on this.

I built a server of my own from scratch some years ago. VMware's hardware compatibility guide stated the motherboard was supported. The hardware manufacturer stated it was all supported. I purchased the board, built my server, used the built-in RAID software to configure a RAID 1 array for my storage drives and ESXi failed to recognize the RAID array. ESXi was still able to use all of the drives, but any attempts to use the RAID software to try and create some form of fault tolerance and data redundancy was met with failure.



13-6: VMware's hardware compatibility list sometimes doesn't tell the whole truth...

If it's so Picky, Why Bother?

Many students may look at the last couple of pages of sidebar discussions and think: "*Why go through all the trouble if it's so picky about the hardware it will actually run on? Is it really worth it?*" Well, there's two reasons why you as a student should put in the effort:

First and foremost, **ESXi is probably one of the most commonly used hypervisors in enterprise networks.** What that means is that a lot of organizations use it, so it makes sense to try and set it up in a lab environment, in order to get more familiar with it. This is hands-on experience for you.

The second reason you should consider putting in the effort is for troubleshooting experience. **Troubleshooting is a difficult skill to acquire, and even harder to master.** Forcing ESXi into submission and getting it to run on your hardware is a task that will test troubleshooting prowess. **Here are some questions that can help guide you in trying to troubleshoot problems that come up:**

What exactly is the problem? Being able to describe the problem and steps taken to resolve it is an extremely valuable technical skill. Effective communication is the key to getting your problems resolved. Make sure to document the problem, and its solutions thoroughly once you have discovered them.

What are my options for solving the problem? The most common problem folks run into trying to install ESXi is either lack of SATA controller drivers, or network drivers. Which is the root of the problem? From there, can you determine which hardware vendor produces the SATA controller or network interface? From there, can you determine whether or not there are third-party drivers available for ESXi? How much would it cost to install compatible hardware?

Which options are feasible given time and budget to dedicate towards this task? Maybe you found a third-party driver for your NIC or SATA controller. Can you perform the steps necessary to add it to the ESXi installation ISO? Maybe you have access to a compatible RAID controller and/or network card. Do you know how to install hardware to your server? If you don't know how to do these things, is there anyone you can rely on to walk you through the necessary steps?

What are my alternatives if I cannot easily resolve this problem? If ESXi isn't working for you, and either you can't find drivers or inject them into the ESXi ISO, and you don't have funds for new hardware, what other options are there? I know it seems contradictory to recommend another bare-metal hypervisor, especially another bare-metal hypervisor not covered by this book (yet) but if the end-goal is getting a working virtual lab up and running, and ESXi isn't cutting it, that is the next best solution.

If you come to a crossroads and decide that ESXi is not worth the time or the effort, here are a couple of other bare-metal hypervisors students can try out:

Hyper-V Server: <https://www.microsoft.com/en-us/evalcenter/evaluate-hyper-v-server-2019>

Windows Server 2019: <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019>

Microsoft provides a stand-alone version of their Hyper-V hypervisor. The good news is that the software is available for free. The bad news is that if you're used to Windows Server, Hyper-V Server is absolutely nothing like it. You get a barebones command-line interface, and are expected to manage the system remotely via the Hyper-V Manager application.

Microsoft also provides evaluation copies of Windows Server that last 180 days. Students can install the Hyper-V server role if they're looking for a graphical option that is a little easier to set up.

Proxmox VE: <https://www.proxmox.com/en/proxmox-ve>

Proxmox is an open-source hypervisor. It's considered much less picky about the hardware it supports, while being fairly easy to use.

13.2 Installing ESXi

Before we begin, students will need to register an account on vmware.com in order to download a copy of ESXi and acquire a license key. Afterwards, they will need to convert installer into a bootable USB drive, using a utility called UNetbootin. Finally, we will use that bootable USB drive to install ESXi onto a second USB drive or SD card.

13.2.1 Acquiring the installation ISO

Open a web browser and navigate to <https://www.vmware.com>. In the upper right corner of the page, there are a series of navigation options. Click the option labeled *Login*, and in the dropdown that appears, click *Customer Connect*. This brings students to a login page. Look for the text, *Don't have an account Sign up now*, and click on the link to register a new *Customer Connect* account. Fill out the registration form, and make sure to provide a valid email address, because in order to log in with your account, VMware will send a confirmation email that students are required to access in order to activate their account. After clicking the *Activate Now* link in that email, re-enter the account password to activate the account, and log in. After a moment or two, students will be redirected to the *Customer Connect* site. From here, enter the URL:

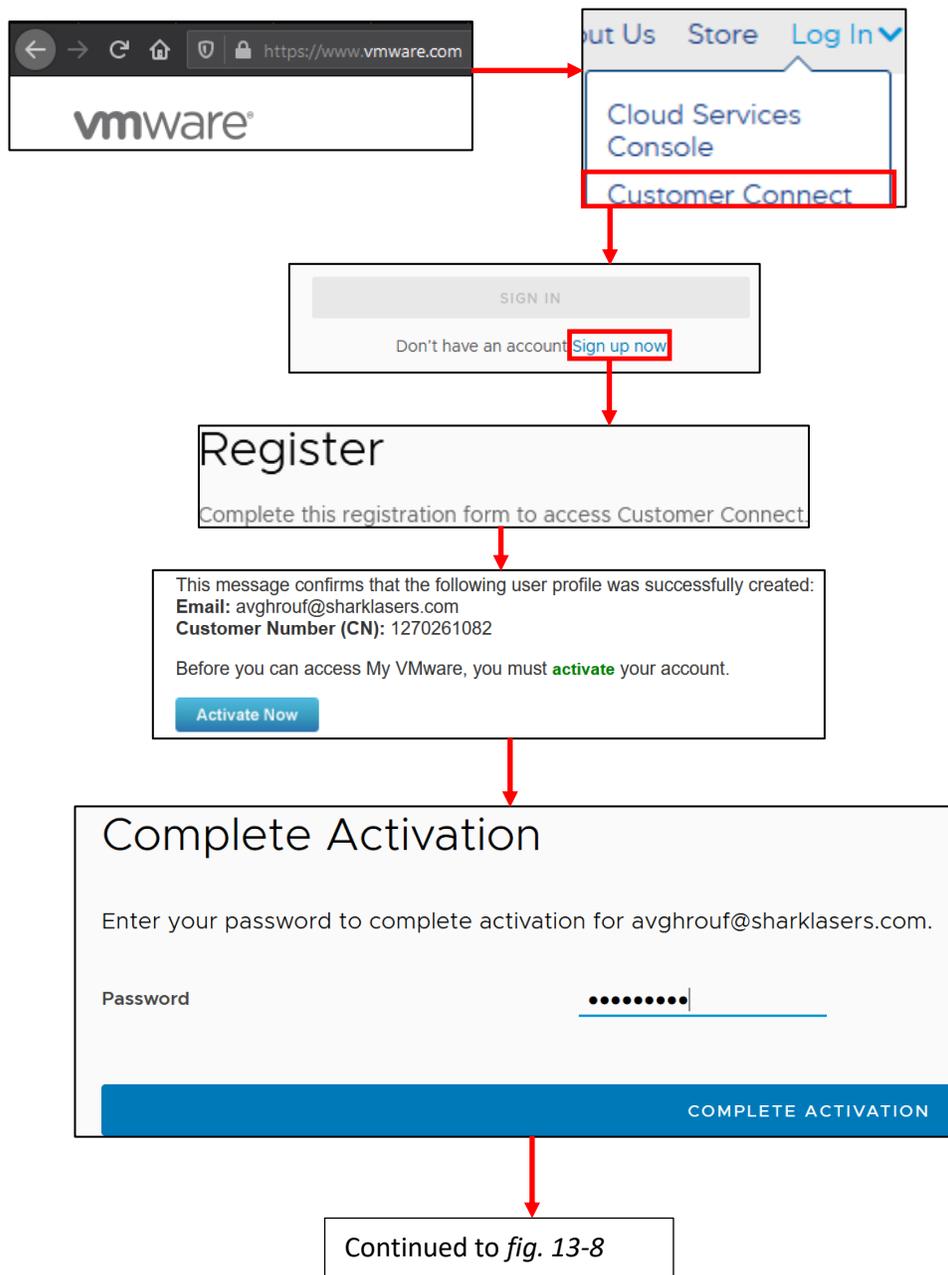
<https://my.vmware.com/en/group/vmware/evalcenter?p=free-esxi7>

And hit enter to proceed. This will take students directly to the VMware vSphere Hypervisor 7 Download Center.

Click the tab labeled *License & Download*, and click the *Register* button below. VMware will ask you to confirm your contact information, and accept the terms of their licensing agreement. Fill out the fields they request and submit that information. You should get redirected back to previous page and under the License and Download tab, new information should appear.

Under the section labeled *License Information*, make sure to copy the License key that is displayed. In fact, **I would highly recommend saving your vmware.com username, password, and ESXi license key to your password manager for safekeeping.** Under the *Download Packages* section, click the *Manually Download* button next to the *VMware vSphere Hypervisor (ESXi ISO) Image* listing.

Note: Like most websites, vmware.com is subject to constant change. Links, URLs, and page layouts may not exactly match the instructions and illustrations I've laid out. Don't panic, remember that software is subject to constant change, and when all else fails, utilize the website's search features, or your favorite search engine to help you find what you are looking for. As always, students should ensure they are downloading the latest and most up to date software available. I recommend utilizing the VMware website's search feature and searching for "vsphere hypervisor" to be directed to the newest version of the hypervisor available.



13-7: Technically ESXi is free, but boy oh boy does VMware make you jump through hoops to get free stuff. First and foremost, register a *Customer Connect* account. There is a lot of information that VMware wants to know about you, but technically, it doesn't have to be *your* contact information. Just make sure you can get the *Activate your account* email in order to click the *Activate Now* link.

Continued from *fig. 13-7*

vsphere hypervisor

<https://my.vmware.com/en/group/vmware/evalcenter?p=free-esxi7>

Installation & Configuration >

License & Download >

Register

Accept End-User License Agreement

To complete your request, check and update your profile, enter any missing information, and accept the End-User License Agreement at the end of the form.

License Information

COMPONENT	LICENSE KEYS
VMware vSphere Hypervisor 7 License	

Download Packages

Your downloads are available below

VMware vSphere Hypervisor - Binaries

VMware vSphere Hypervisor (ESXi ISO) image
2020-06-23 | 7.0b | 351.9 MB | iso

Manually Download

13-8: After logging in, navigate back to vmware.com. I recommend using the VMware website's search feature to search for "vsphere hypervisor" in order to be directed to the download page for the latest version of vSphere Hypervisor (ESXi), but for students looking for a direct link to vSphere hypervisor 7.0, navigate to:

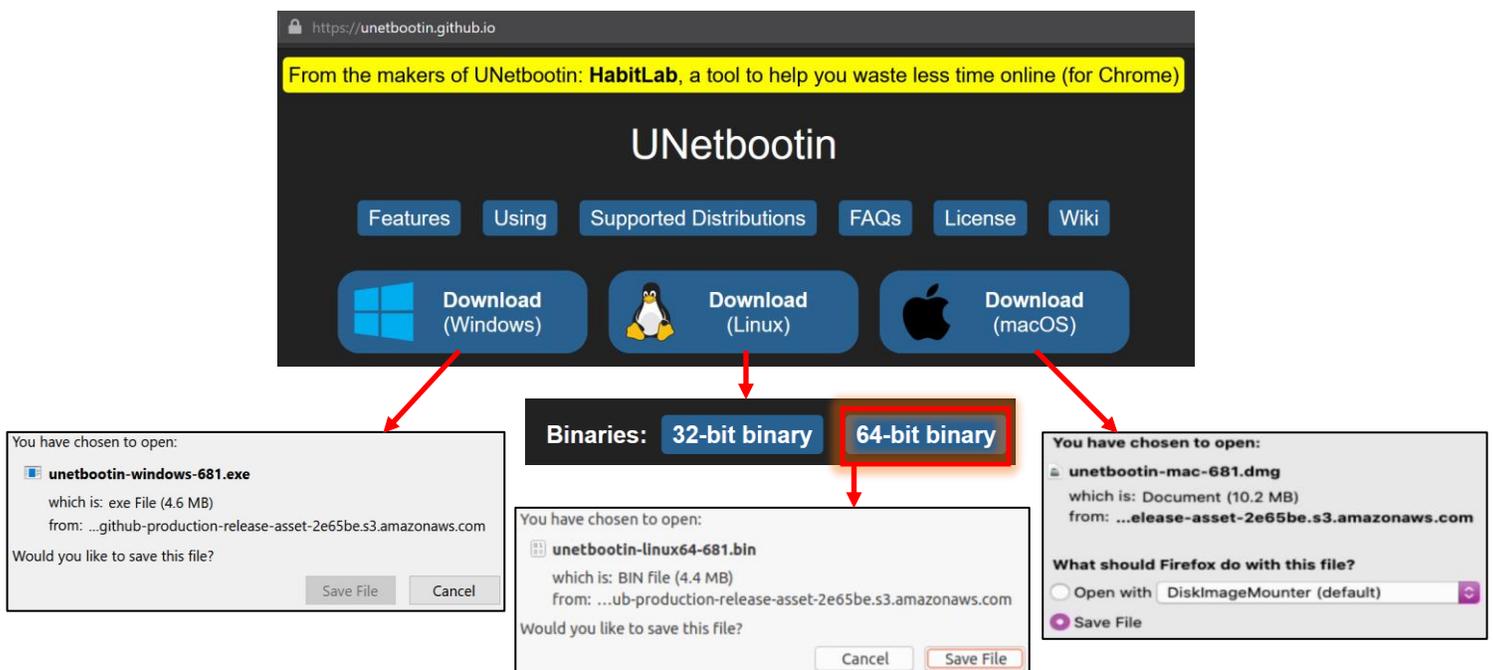
<https://my.vmware.com/en/group/vmware/evalcenter?p=free-esxi7>

Click the *Register* button under the *License & Download* tab, and VMware asks you to accept an end-user license agreement. After accepting the license agreement, you are sent back to the previous page, but now the *License & Download* section features the *License Information* and *Download Packages* section. Record the entry under the *License Keys* section, and click the *Manually Download* button beside the *VMware vSphere Hypervisor (ESXi ISO) image* listing. **Save your username and password to vmware.com and the ESXi license key to your password manager!**

13.2.2 Downloading and Installing UNetbootin

With the ESXi ISO downloaded, the next step is to create a bootable USB drive with the installation media. We'll be using a special utility called UNetbootin to do this. Most operating systems are distributed as CD image files known as ISO files – just like the ESXi ISO we just acquired. In the before times, users would burn these ISO files to a CD or DVD, and use that to install an operating system of their choosing. This utility reads the ISO file, copies the installation files to a USB drive, then performs the necessary incantations to make that USB drive bootable, so that the USB drive can be used instead of an optical drive. There are many other utilities that do what UNetbootin does, but the nice thing about it UNetbootin in particular, is that it is multiplatform. This means that UNetbootin runs on Windows, Linux, and MacOS.

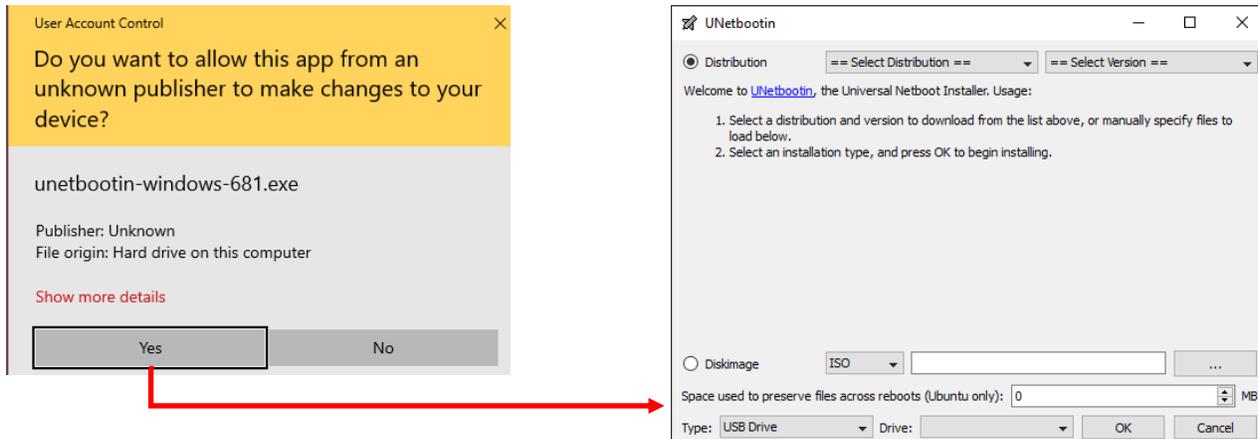
To begin, open your preferred web browser, and navigate to <https://unetbootin.github.io/>. Click the download link for the operating system you are using. MacOS and Windows users get a direct link to an installer executable (Windows), or an installation DMG (MacOS). However, the Linux users are given a choice of attempting acquire the tool from their distribution's software package manager of choice, or downloading a 32-bit or 64-bit bin file. For ease of use, I'm going to recommend downloading one of the bin files, as opposed to mucking around with package managers. I'm going to make an assumption here and guess that if a student reading this book is brave enough to be using Linux as a desktop operating system, they'll be able to determine whether or not they need the 32-bit or 64-bit bin file. When in doubt however, most modern Linux distributions are 64-bit by defaults these days, so the 64-bit bin file is recommended.



13-9: Make your way to unetbootin.github.io to get started. Windows and MacOS users get direct links to installer files, while Linux users get various options for acquiring the software. To keep things simple, downloading one of the bin files is recommended. When in doubt, download the 64-bit bin file.

13.2.2.1 Installing UNetbootin on Windows

Download the Windows executable, and double click to run it. If UAC is enabled, students may be presented with a UAC prompt asking if they want to allow the software to make changes. Select *Yes* to continue, and the application will open. That's all there is to it. UNetbootin is a PE (portable executable), and can be run from practically any directory.



13-10: Installation on Windows is pretty straightforward. If students get a UAC prompt when running the executable, select *Yes* to continue.

13.2.2.2 Installing UNetbootin on MacOS

Download and open the DMG file by double clicking on it. A folder opens with a readme file and the UNetbootin executable. I recommend copying and pasting the executable to your desktop, then double clicking on it to run it. The first time running the application, students will be prompted to confirm they wish to run the application. Click the *Open* button to continue. If students are running MacOS Catalina (10.15) or later, they may get additional prompts indicating that the application wishes to access the desktop. Click *OK* to continue. Then, students will be prompted to enter their password to provide the application permission to write files to connected USB drives. Enter the MacOS account password, then click *OK* to continue to the application.



13-11: Download the UNetbootin DMG file, then double click on it to open it. Copy the executable to the desktop, then double-click to open it. Students should prepare themselves for the litany of permission requests the operating system will prompt them with, when attempting to do anything.

UAC, Except Somehow Worse

Somewhere between the *Mac. Vs. PC* commercials of the early 2000s and MacOS Catalina (10.15+), Apple somehow turned MacOS into the very operating system they were criticizing when Windows Vista came out. At the time, everyone criticized Windows Vista because of how annoying and invasive User Account Control (UAC) was, with its seemingly endless prompts to confirm users wanted to perform some task or another. Fast-forward about decade, and you probably noticed this annoying behavior when downloading and running UNetbootin for the first time on MacOS. It's not enough to confirm that you want to install the application anymore, you must also allow the application permission to perform various tasks.

On the one hand, it's very much like how when you download a smartphone app, you have to give the application permission to do things like read your contacts, access the filesystem, etc. This permissions system is designed to protect the end-user from programs that attempt to do sketchy things to your system. On the other hand, it's a very tedious form of *mother may I*. You download an app, you double-click to run it, you have to tell MacOS *yes, I know what I downloaded* and *yes, I want to run it*. Then you have to allow the application permissions as it requests them. This creates a condition called *alert fatigue* and causes most users to just click *OK* to continue, because that's how they get rid of the prompt that is interrupting their work. This defeats the entire purpose of the security feature because it's so annoying and invasive, that users no longer care what it has to say. Unfortunately, if you're an MacOS user, you're going to have to deal with it for now. Hopefully Cupertino will see fit to improve this security feature, and make it a little less invasive and annoying in the future.

13.2.2.3 Installing UNetbootin on Linux

UNetbootin has a software dependencies that are required in order to run on Linux. Students will need to download and install the software packages p7zip and syslinux. Different Linux distributions have different methods, and different names for the p7zip package:

Debian-based distro users (e.g. Ubuntu, Debian, etc.): use the apt package manager to download and install the packages p7zip-full, and syslinux using the command:

```
sudo apt-get install p7zip-full syslinux
```

Redhat-based distro users (e.g. Redhat Enterprise Linux, CentOS, etc.): Use the dnf package manager (RHEL/CentOS 8, etc.) and install the packages p7zip, p7zip-plugins, and syslinux using the command:

```
sudo dnf install p7zip p7zip-plugins syslinux
```

Note: You'll need to enable the EPEL software repositories for CentOS or Redhat in order to be able to download the p7zip or p7zip-plugins packages. For RHEL/CentOS 8 this is as simple as running:

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

Please note that these instructions are specific to CentOS and Redhat 8 users. If you're using Redhat/CentOS 7 or older, you'll want to use the yum package manager, and install the EPEL software repository for the version of Redhat/CentOS you are running instead.

```
[ayy@localhost ~]$ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-r
epel-release-latest-8.noarch.rpm
[sudo] password for ayy:
Last metadata expiration check: 0:51:53 ago on Mon 07 Sep 2020 08:02:21 PM EDT.
epel-release-latest-8.noarch.rpm          67 kB/s | 22 kB    00:00
Dependencies resolved.
=====
Package                Architecture  Version      Repository      Size
=====
Installing:
epel-release            noarch       8-8.el8      @commandline    22 k
Transaction Summary
=====
Install 1 Package

Total size: 22 k
Installed size: 32 k
Is this ok [y/N]: █
```

13-12: Redhat and CentOS users **must** enable the EPEL repos to acquire the p7zip software packages.

Other Linux distros: I'm going to assume that if students are brave enough to be using more exotic Linux distributions as their desktop operating system (e.g., Arch, Gentoo, or even more exotic stuff), that they are either familiar enough with the package manager for the distribution to find the correct software packages to install, or familiar enough with compiling software from source to acquire p7zip and/or syslinux their own.

```
[ayy@localhost ~]$ sudo dnf install p7zip p7zip-plugins syslinux
[sudo] password for ayy:
Last metadata expiration check: 0:48:41 ago on Thu 10 Sep 2020 12:04:23 PM EDT.
Dependencies resolved.
=====
Package                Architecture  Version      Repository    Size
=====
Installing:
syslinux                x86_64       6.04-4.el8   BaseOS        579 k
p7zip                   x86_64       16.02-16.el8 epel           691 k
p7zip-plugins           x86_64       16.02-16.el8 epel           1.1 M
Installing dependencies:
syslinux-nonlinux      noarch       6.04-4.el8   BaseOS        552 k
=====
Transaction Summary
=====
Install 4 Packages

Total download size: 2.8 M
Installed size: 8.6 M
Is this ok [y/N]:
```

```
ayy@ayy:~/Downloads$ sudo apt-get install p7zip-full syslinux
[sudo] password for ayy:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  p7zip syslinux-common
Suggested packages:
  p7zip-rar
The following NEW packages will be installed:
  p7zip p7zip-full syslinux syslinux-common
0 upgraded, 4 newly installed, 0 to remove and 205 not upgraded.
Need to get 1,349 kB/2,894 kB of archives.
After this operation, 9,971 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

13-13: UNetbootin on Linux requires the software packages p7zip and syslinux. On Ubuntu/Debian-based systems, students will need to use the apt package manager to download p7zip-full and syslinux, while Redhat/CentOS users will need to enable the EPEL software repository, and use the dnf package manager to acquire p7zip, p7zip-plugins, and syslinux.

With prerequisites out of the way, Download the 64-bit UNetbootin binary for Linux. Open a terminal window, and use the cd command to change directories to where the binary was downloaded. Usually this will be in a folder called "Downloads" in the current user's home directory, so run:

```
cd ~/Downloads
ls -al
```

"~/ " is a short-cut that tells the cd command to "Change directories to the current user's home directory". For example, if my username was ayy1mao, the home directory would be /home/ayy1mao (usually). When students use this shortcut, this should place them in the directory /home/ayy1mao/Downloads.

Students can run the command `pwd` to print the name of the current directory and see this for themselves. `ls -al` will list out the contents of the `Downloads` directory. Using this, students can confirm that the UNetbootin binary is present. Next, students will need to run the command:

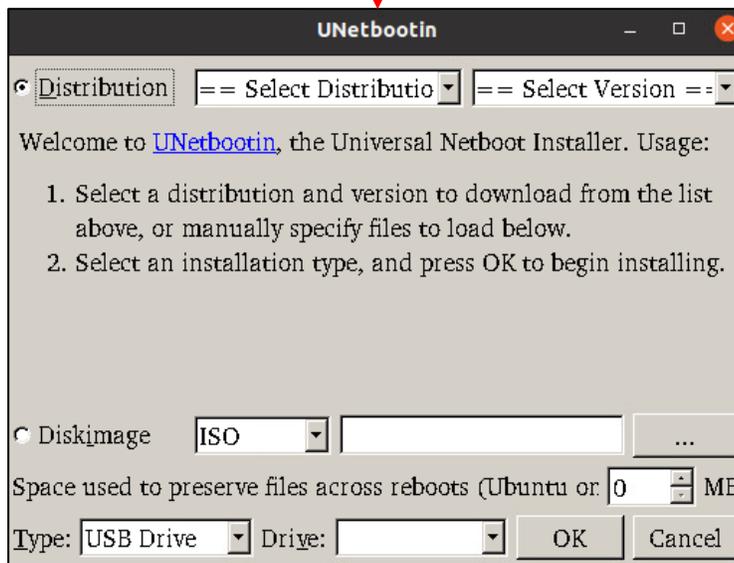
```
chmod u+x unetbootin-linux64-681.bin
```

This command changes the file permissions of the file and provides us permissions to e(x)ecute the file. Finally, students will need to run `unetbootin` as root using `sudo`:

```
sudo ./unetbootin-linux64-681.bin
```

Please note that as of writing this, version 681 was the most recent version of UNetbootin, and that portion of the filename may change.

```
ayy@ayy:~$ cd ~/Downloads/  
ayy@ayy:~/Downloads$ chmod u+x unetbootin-linux64-681.bin  
ayy@ayy:~/Downloads$ sudo ./unetbootin-linux64-681.bin  
[sudo] password for ayy: █
```



13-14: Download the UNetbootin binary, `cd` to the `Downloads` directory, change the file permissions so that the file can be ran/executed, then use `sudo` to run the `unetbootin` bin file.

Special Incantations for Redhat/CentOS Users

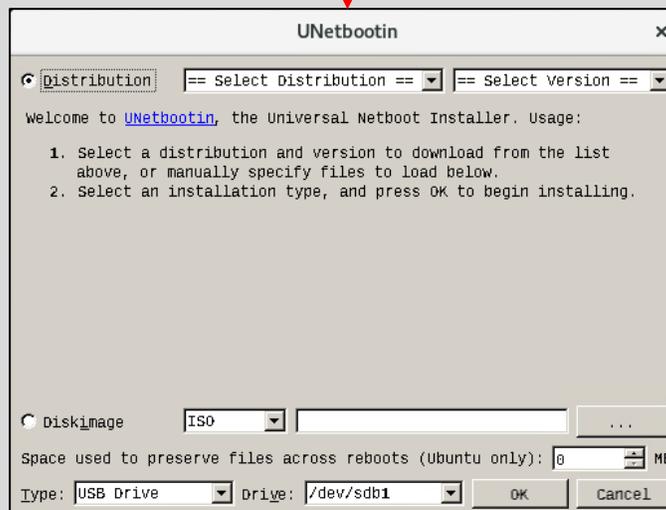
Redhat and by extension CentOS are about as cutting-edge as a dull soup spoon. Sometimes this means that there are some unique problems that they run into when running certain utilities. Seems that UNetbootin is among those utilities. In order to run UNetbootin on RHEL/CentOS, you need to run the following commands after the chmod command listed above:

```
xhost local:root
sudo QT_X11_NO_MITSHM=1 ./unetbootin-linux64-681.bin
```

If everything was entered correctly, the application should start. These instructions were taken from the unetbootin github:

<https://github.com/unetbootin/unetbootin/issues/94>

```
[ayy@localhost ~]$ cd ~/Downloads/
[ayy@localhost Downloads]$ chmod u+x unetbootin-linux64-681.bin
[ayy@localhost Downloads]$ xhost local:root
non-network local connections being added to access control list
[ayy@localhost Downloads]$ sudo QT_X11_NO_MITSHM=1 ./unetbootin-linux64-681.bin
[sudo] password for ayy:
```



13-15: Just like with Ubuntu/Debian hosts, cd to the Downloads directory and change the file permissions of the UNetbootin binary. Then, run the xhost local:root command. This command controls access to the local window manager called the "X" server. Finally, our last command, sudo QT_X11_NO_MITSHM=1 ./unetbootin-linux64-681.bin calls the sudo command to both set a special environment variable (QT_X11_NO_MITSHM=1), and execute the UNetbootin binary as the root user.

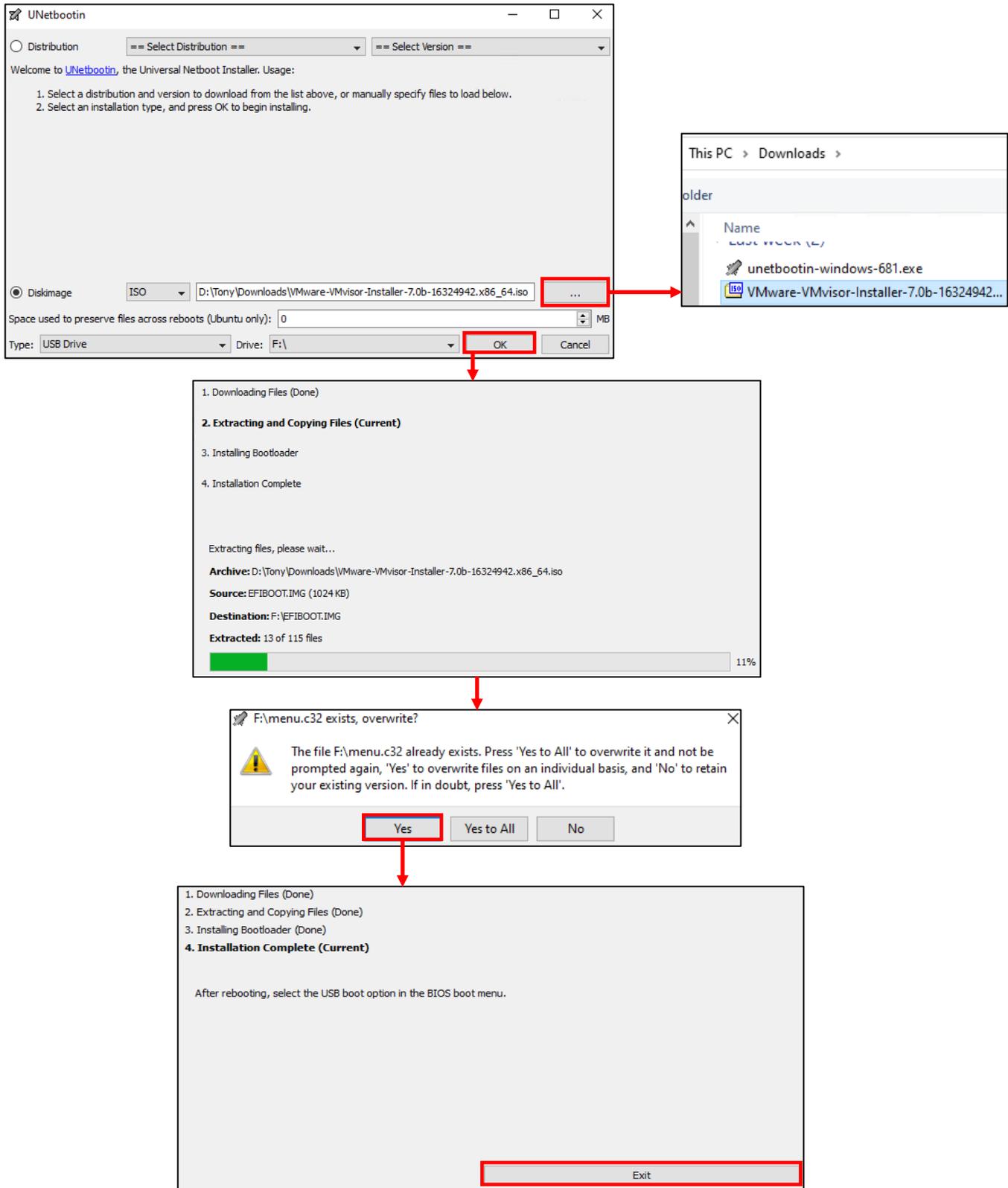
13.2.3 Using UNetbootin to create a bootable installer USB drive

Fortunately, UNetbootin's interface is pretty consistent across Windows, Linux, and MacOS. That means creating your bootable USB drive will be a pretty easy task, and the steps will be easy to follow, no matter what desktop operating systems students use. **Students should insert the USB drive they wish to use for holding the ESXi installer image before proceeding.**

Open the UNetbootin application, and click on the radio button labeled *Diskimage*. Next to this radio button, there should be a drop-down menu. The default option should be *ISO*. If not, please make sure *ISO* is selected. Finally, next to that drop-down box is an input box, and a button labeled with three dots (...) click on this button to open your operating system's file browser, and browse to the location of the ESXi hypervisor ISO (The name of this file for ESXi 7 should be *VMware-VMvisor-Installer-7.0b-16324942.x86-64.iso*. Remember that downloaded files are usually in the `Downloads` directory of the currently logged in user). Select the file, and the blank input box updates to point towards the location of this ISO file.

Next, look at the drop-down labeled *Type*. By default, the option *USB Drive* should be selected, but if it is not, be sure to select it. Right beside the *Type* field, is the *Drive* field. If students only have a single USB drive plugged in, then UNetbootin is smart enough to find and automatically select the correct disk device (Linux, MacOS) or drive letter (Windows). Otherwise, you'll need to make sure you are select the correct disk, because **UNetbootin will delete all of the files on the thumb drive you use to store the ESXi installation ISO.**

When ready, click the *OK* button to continue. UNetbootin will then proceed to configure the USB drive, and copy the installation files from the ISO file students downloaded earlier. A pop-up titled, *menu.c32 exists, overwrite?* may appear. If this pop-up appears, click *Yes* to continue. When finished, the text **4. Installation complete (Current)** will appear in bold text, along with the notification: "*After rebooting, select the USB boot option in the BIOS boot menu.*" Click the *Exit* button to close UNetbootin, and remove the USB drive.



13-16: Before starting UNetbootin, please make sure the USB drive is inserted. Select the *Diskimage* radio button, the *ISO* option in the drop-down to the right, then click the ellipsis (...) button to browse to the ESXi ISO. Down below, make sure *USB Drive* is selected as the *Type* and that the correct drive letter (Windows) or disk device (MacOS, Linux) is selected, then press *OK*. If prompted to overwrite the *menu.c32* file, select *Yes*, or *Yes to All* to proceed. Finally, when the process is finished, Click the *Exit* button and remove your bootable USB drive.

As Finished as A Half-Eaten Sandwich

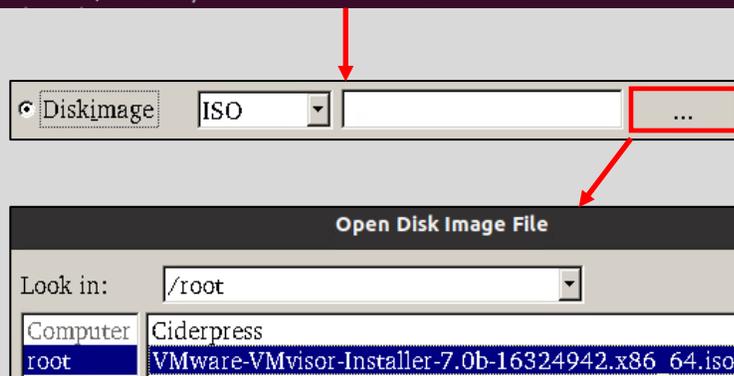
In addition to the pre-requisites and special incantations required to get UNetbootin to actually run on Linux, it appears that some of the functionality doesn't work terribly well. Clicking the ellipsis button (...) will load a file browser window, but from my experiences on both Ubuntu Desktop 20.04 and CentOS 8, there is no ability to actually browse the file system. Users can only select files located in the "/root" directory – the root user's home directory.

This obstacle can be overcome by copying the ESXi ISO to the /root directory. This can be done using the cp command:

```
sudo cp ~/Downloads/VMware-VMvisor-Installer-7.0b-16324942.x86-64.iso /root
```

You may need to adjust the directory path depending on where you downloaded the ESXi ISO to. You might also need to adjust the filename if you're using a different/newer version of ESXi than what I had available at the time. Remember that you can start typing the name of the file and hit tab, and most shell environments will complete the rest of the filename for you (tab completion). After running this command, restart UNetbootin, and the ISO should be available under the /root directory. Select it, and continue creating the USB installation drive. If this is not an option for one reason or another, unfortunately, the only recommendation I can make is to run UNetbootin on a Windows or MacOS system (or virtual machine) and use that to create your USB drive, instead.

```
ayy@ayy:~/Downloads$ sudo cp ~/Downloads/VMware-VMvisor-Installer-7.0b-16324942.x86_64.iso /root
[ayy@ayy:~/Downloads$ sudo ./unetbootin-linux64-681.bin
```



13-17: For one reason or another, UNetbootin on Linux has problems letting users browse to other directories. Saavy users will notice the error `QPixmap::scaleWidth: QPixmap is a null pixmap` in their terminal app any time they attempt to browse to another folder. Apparently, this has been a known since version 613 (<https://bugs.launchpad.net/unetbootin/+bug/1482292>). The work-around is to copy the ESXi ISO file to the /root directory so that users can select it from there, as illustrated above.

13.3: Installing ESXi

Now that students have a bootable USB drive, the next step is to install ESXi on the server they'll be using host the hypervisor. **As a reminder, students will need two USB drives, or a USB drive, and SD card (and an SD card slot on the server, of course). One USB drive to host the installation files, and the other USB drive (or SD card) that is at least 8GB in size for us to install the ESXi hypervisor.**

The next step is to configure the server to boot from the USB drive you prepared using UNetbootin. It's very common for most systems to use either the F1, F2 or DEL key to enter the system BIOS while the system is booting. Once in the BIOS menu, usually the option that controls which device(s) and what order to check them for a bootable operating system will be referred to as the *Boot Order*. Students will need to consult system and/or motherboard documentation to determine how to access the system BIOS and configure their system to boot from the USB drive.

If you did everything correctly, students will be greeted with a grey window, titled *Welcome to the VMware ESXi 7.0.0 Installation*. Hit the enter key on your keyboard to continue. On the next screen, students are presented with an End User License Agreement (EULA), hit the F11 key on your keyboard to proceed. After a moment or two, ESXi will present you with a screen that contains all of the storage devices it was able to detect. On my system, the USB drives were identified as General UDisk. There were two of them, nearly identical with the exception of a single-digit difference in the hardware identifier. If students are utilizing two identical USB drives – one containing the ESXi installer, and another in which they are going to install ESXi to, and are having a hard time differentiating which is which, here is the most effective method I have found to tell them apart. **Follow these instructions exactly:**

-Disconnect the USB drive that has the ESXi installation files. Trust me on this. Remember what USB port the drive was plugged into.

-With the thumb drive removed, hit the F5 key to force the ESXi installer to rescan the available storage devices. This should cause one of the USB drives to disappear. Use the arrow keys to highlight the remaining USB drive.

-Before hitting enter to continue, plug the installation USB drive back in to the same USB port it was plugged into initially. Wait 10 seconds, then hit the Enter key to continue.

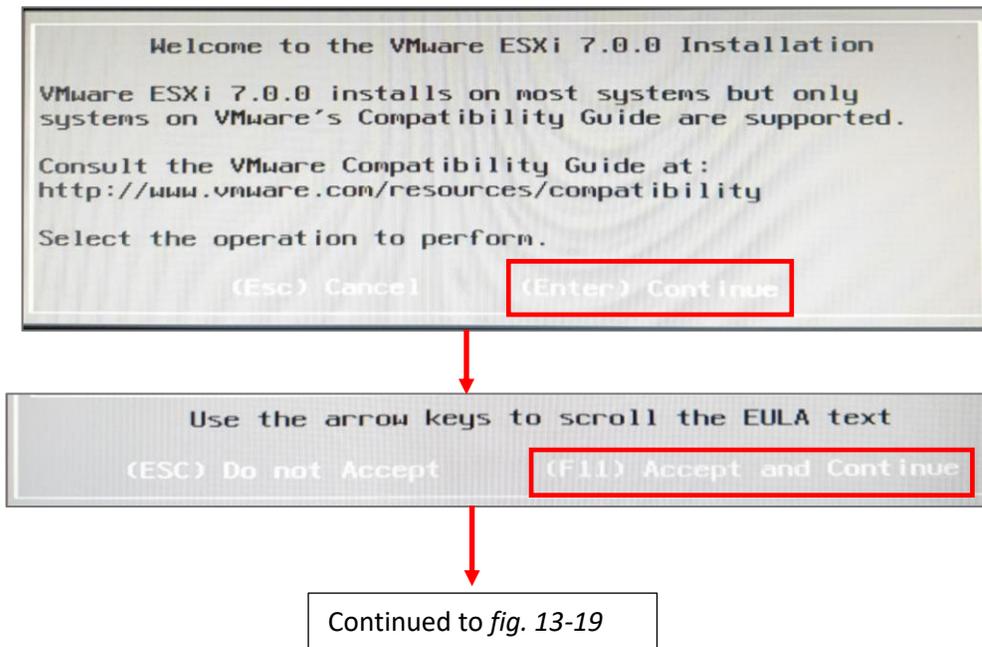
A pop-up appears, labeled *Confirm Disk Selection* will appear. This pop-up notifies users that the data on the disk selected will be overwritten to install ESXi. Hit the Enter key to proceed.

The next screen asks users to select a keyboard layout. The default setting is *US Default*. If students require an alternate keyboard layout, use the arrow keys on the keyboard to highlight the correct keyboard layout. Press the Enter key to continue.

Next up, users are asked to enter a root password, then confirm it. These will be the credentials students use to log in to ESXi from either the local console and/or the web-based interface momentarily. **The installer requires that the password consists of at least 7 characters, At least 1 capital letter, 1 number, and 1 special character.** Use the arrow keys to transition between the Root password and Confirm password input boxes. Once finished, press the Enter key.

The installer will scan the system hardware once more to confirm it is all supported and functioning correctly. If presented with a warning regarding hardware support, press the Enter key to proceed. Students will get one last dialogue prompt to confirm they want to install ESXi on their thumb drive. Hit F11 to confirm and proceed with the installation.

After some time passes, the installer will display a final screen stating the installation is complete. Remove the installer USB drive, and hit the Enter key once more to reboot the system.



13-18: Welcome to the ESXi installer. Please pardon the potato-quality photos I've taken of this process with my potato-quality cell phone. Some of the white text on grey background may be a little difficult to discern. The first screen students should see upon successfully booting from the installer thumb drive is the welcome screen. Hit Enter, and immediately smash the F11 key to accept the EULA without reading it, as is tradition.

Continued from fig. 13-18

```
Storage Device
-----
Local:
* ATA      WDC WD5003ABYX-0 (t10.ATA_____WDC_WD5003
* ATA      Corsair Force 3 (t10.ATA_____Corsair_Fo
* ATA      WDC WD5003ABYX-0 (t10.ATA_____WDC_WD5003
  ATA      ST1000DM003-9YN1 (t10.ATA_____ST1000DM00
General  UDisk          (mpx.vmhba32:C0:T0:L0)
General  UDisk          (mpx.vmhba33:C0:T0:L0)

(Esc) Cancel  (F1) Details  (F5) Refresh
```

```
Storage Device
-----
Local:
* ATA      WDC WD5003ABYX-0 (t10.ATA_____WDC_WD5003ABY...)
* ATA      Corsair Force 3 (t10.ATA_____Corsair_Force...)
* ATA      WDC WD5003ABYX-0 (t10.ATA_____WDC_WD5003ABY...)
  ATA      ST1000DM003-9YN1 (t10.ATA_____ST1000DM0032D...)
General  UDisk          (mpx.vmhba33:C0:T0:L0)
Remote:

(Esc) Cancel  (F1) Details  (F5) Refresh  (Enter) C
```

```
Confirm Disk Selection

You have selected a disk that contains at least one
partition with existing data.

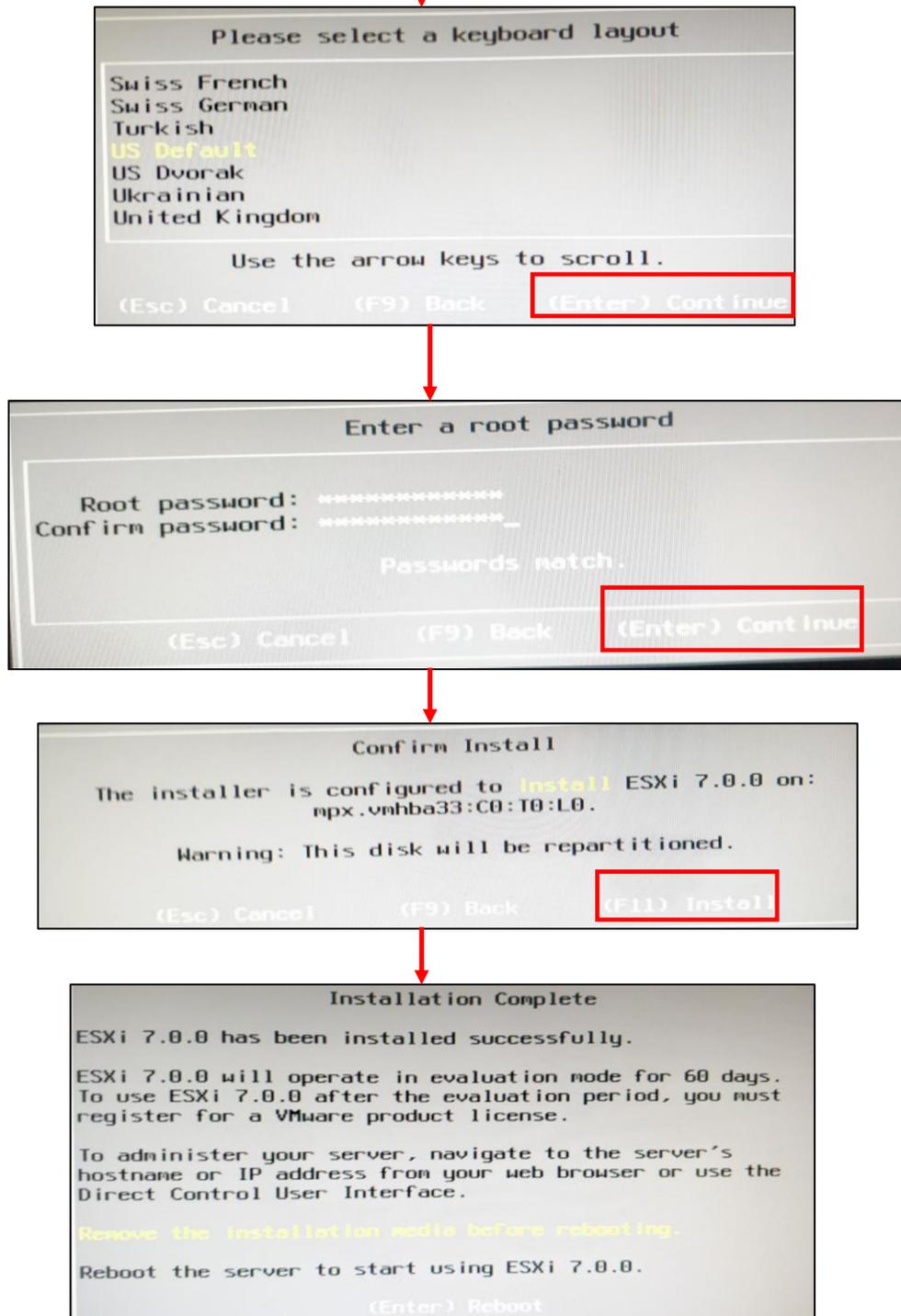
If you continue the selected disk will be overwritten.

(Esc) Cancel  (Enter) OK
```

Continue to fig. 13-20

13-19: Provided everything is going well and the server hardware is compatible, the installer will look for storage drives so that ESXi can be installed. Notice the two devices listed, labeled General UDisk (vmhba32 and 33). ***There's no easy way to tell which of these is the installer USB drive, and the target installation drive. The best method to tell the two apart, is to unplug the installer USB drive, hit F5 to cause the installer to rescan the attached storage devices, highlight the remaining USB drive that is present on the list, plug the installer USB drive back into the same USB port it was unplugged from, then hit Enter to continue.*** If done correctly, a pop-up will appear to confirm that students are aware that installing ESXi will overwrite any data in the USB drive. Press the Enter key to proceed.

Continued from *fig. 13-19*



13-20: The remainder of the installer is pretty straightforward. Select the appropriate keyboard layout, set a password for the root user that meets complexity requirements, then confirm perform the installation. Once finished, remove the installer USB drive, and reboot the server by pressing Enter.

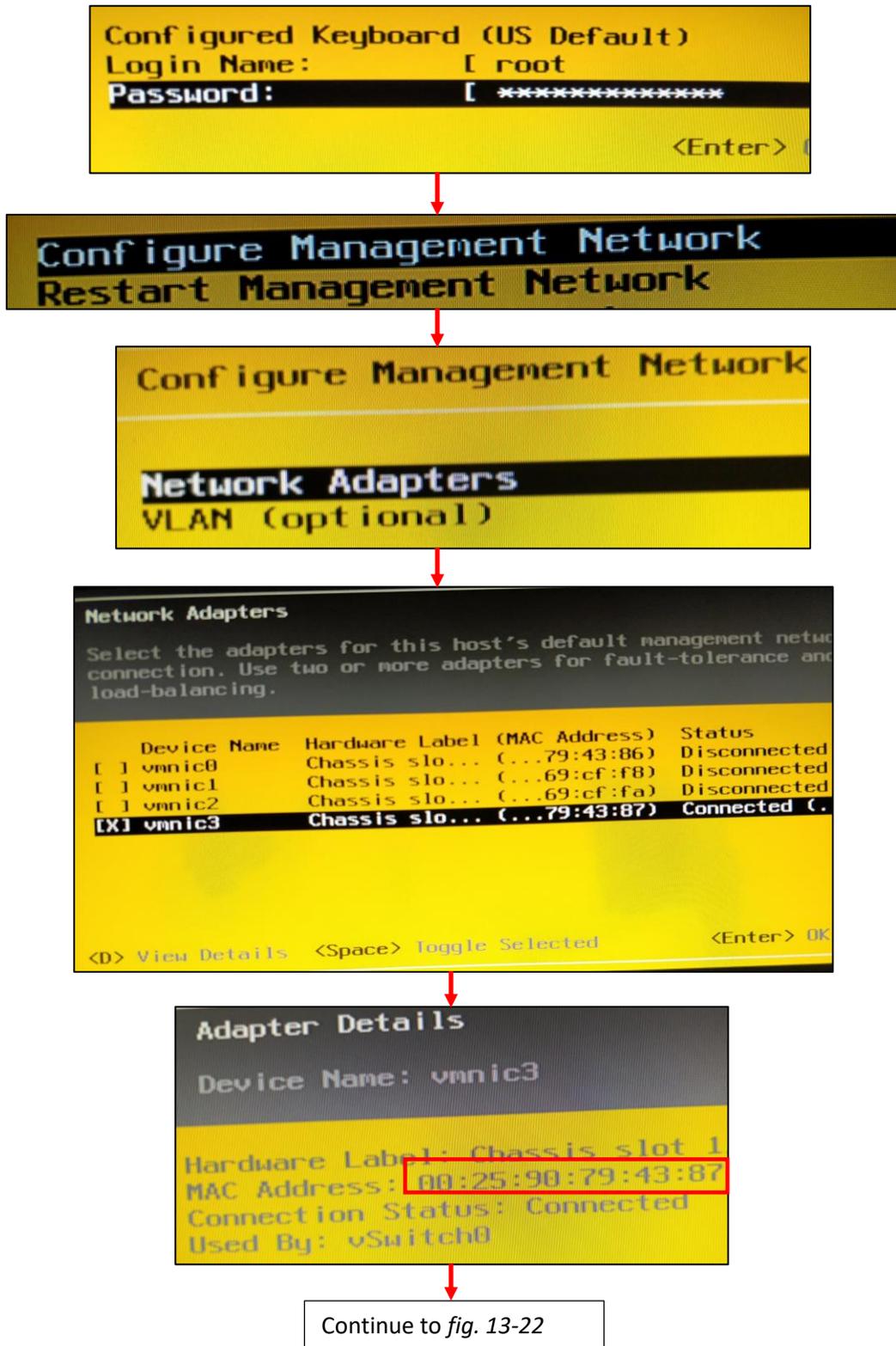
13.4: Accessing the ESXi Web Interface

When the server finishes rebooting, students are greeted with a grey and white screen instructing them to visit one of a series of web addresses in order to manage the ESXi server. Before doing this, students should ensure that the IP address assigned is either statically configured, or provided by a static DHCP mapping. To make a long story short, a lot of networks use DHCP to give IP addresses and network connectivity automatically to newly connected systems. A static DHCP mapping ensures that a system gets the same IP address from the DHCP service. Unfortunately, I have no idea what a student's local network will look like, nor what hardware/software (if any) provides DHCP for their network. What I can do is show you how to recover the MAC address for the management interface of the ESXi server, so that can be used to create a static DHCP mapping.

13.4.1: Configuring a Static DHCP Mapping for the ESXi Management Interface

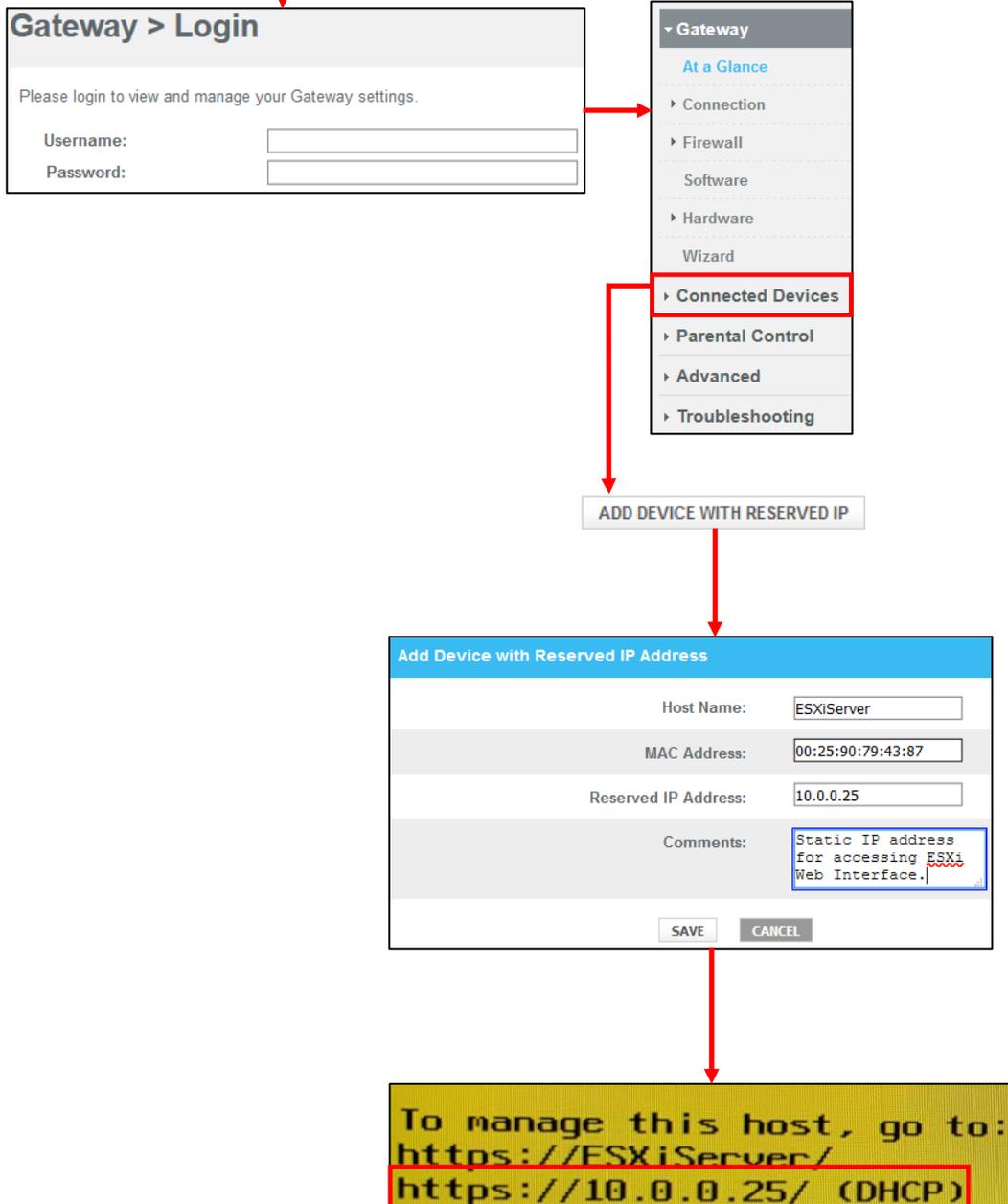
On the lower left portion of the screen, there is an option labeled *<F2> Customize System/View Logs*. Press the F2 key, and ESXi will prompt students to enter the root password to proceed. On the next screen, use the arrow keys to highlight the option *Configure Management Network*, and press Enter. On the next screen, highlight *Network Adapters*, and press Enter again to proceed. ESXi will display a list of all the network adapters on the server it was able to find drivers for. To find the *Management* network interface, look for the *Device Name* that has an X beside it, and a *Status* of *Connected*. Use the arrow keys to highlight that interface then press the D key. A window pops up, labeled *Adapter Details*. Pay attention to the field labeled *MAC Address*, and document the MAC address of this network interface. Once completed, students can repeatedly hit the escape key to navigate back to the main screen.

With the MAC address, students can create a static DHCP mapping. Depending on where the lab is set up, this may be a task that students are responsible for, or a task in which they will have to work with network/system administrators to perform. For students who are creating a virtual lab at home, or for a home office, more than likely your Internet Service Provider (ISP) provides you with some sort of a router/modem combination that provides DHCP service for your home network. Every ISP and vendor has a different way of configuring static DHCP allocations. For demonstration purposes, I will show students the process for creating a static DHCP allocation on the router my ISP provided (fig. 13-22). Begin by logging into the web interface. The navigation menu will likely have the option to display connected devices, or configure the DHCP scope. Click the option to add a reservation, and fill out the input boxes. The IP address range for my home network is 10.0.0.0/24, so I choose to allocate the IP address 10.0.0.25 to my ESXi server. Once finished, save your settings. After creating a static DHCP mapping, the easiest way to make sure it worked, would be to reboot the ESXi server, and confirm it gets the IP address allocated to it by checking the IP address under the *To manage this host, go to* section.



13-21: Hit the F2 key to log into the ESXi console. Select the *Configure Management Network* option, followed by the *Network Adapters* option. Look for the interface with an X beside the *Device Name*, and a *Status* of *Connected*. This is the network interface students will use to connect to the web interface and manage the ESXi server. Highlight it, and the D key to bring up details on the network interface. Document the MAC address.

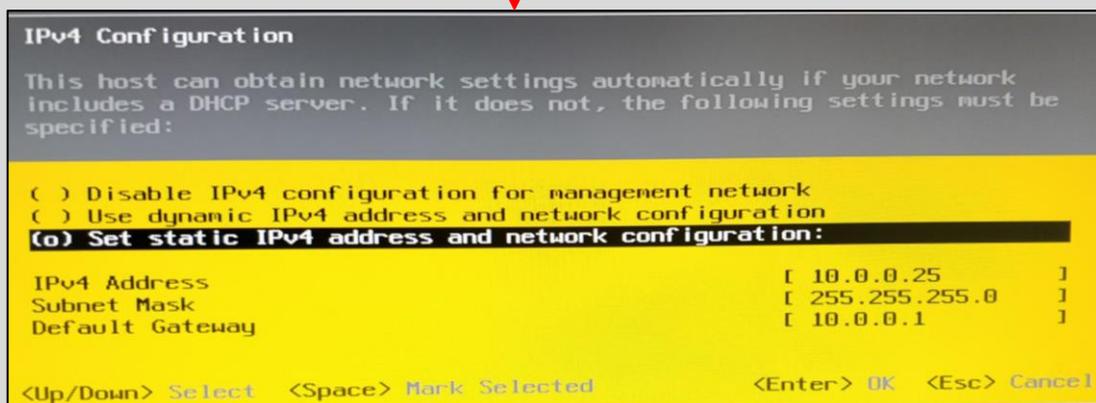
Continued from *fig. 13-21*



13-22: Now that students have a MAC address, they can use that to add a static DHCP mapping for their ESXi server. For students creating a virtual lab on an enterprise network, they may have to work with their network or system administrators to get a static DHCP mapping for their server. For students creating a lab environment at home, it's usually a matter of connecting to the router/modem, and navigating to the correct options on the web interface to configure a mapping. Unfortunately, every small office/home office router is different, so the illustration above may not exactly match the steps students are required to take in order to make a static DHCP mapping on their network equipment. Have the documentation nearby if necessary. ***It's extremely important for the ESXi server management interface to get the same IP address consistently in order to ensure consistent access to the web interface.***

What if I don't Have DHCP?

Some students may be operating on a network where DHCP service isn't provided. Fortunately, ESXi allows users to set a static IP address for the management interface. Log on to the ESXi console by hitting the F2. Select the Configure Management Network option. Use the arrow keys to select the IPv4 Configuration option, and hit enter. The IPv4 Configuration screen appears. Use the arrow keys to highlight Set static IPv4 address and network configuration, then hit the space bar. Next, use the arrow keys to highlight the IPv4 Address, Subnet Mask, and Default Gateway fields, and modify them as necessary.



13-23: From the *Configure Management Network* screen, Select *IPv4 Configuration*, Select the *Set static IPv4 address and network configuration* option, then highlight and change the *IPv4 Address*, *Subnet Mask*, and *Default Gateway* fields as necessary.

Keeping things Managed

You only need one management network interface for your ESXi server, but I highly recommend having at least two network interfaces connected, if at all possible. Only the management network interface needs a static IP address or static DHCP allocation. The *Network Adapters* screen (fig. 13-21, second image from the bottom) allows students to define which network interface is the management network interface. Highlight the desired interface, and press the space bar to add the X next to it (make it the management interface) or to remove the X from it.

Static Cling

While I'm here talking to you about how important it is to ensure the ESXi Management network interface has a consistent, statically configured IP address, **it is also going to be extremely important that the workstation you plan on using to manage your ESXi server, and the WAN interface of the pfSense virtual machine also have consistent, static IP addresses.** We'll talk more about the pfSense VM a little bit later in this chapter. For now, make sure that the system you'll be using to access the ESXi web interface, and manage your virtual machines has a static IP address – either via static DHCP allocation or manually setting its IP address. It's very important that the management workstation's IP address does not change.

Some of you may be setting up your lab environment in a corporate office and it might not be possible or feasible to set a static IP address or DHCP allocation for your workstation.

Fortunately, there's a work-around to this called a "bastion host" or as its more commonly referred to, a "jump box". **Check out Chapter 16, Routing and Remote Access for Bare-Metal Hypervisors, starting on page 835.**

Just so it's absolutely clear, students will need to configure a minimum of three static IP addresses on their local network to maintain consistent access to the ESXi lab environment and virtual machines:

1. The management workstation, or a jump box/bastion host (covered in Chapter 16)
2. The ESXi server's management interface (Covered in [section 13.4](#), pp. 543-549)
3. The WAN interface of the pfSense virtual machine (Covered in [section 13.6.3.1](#), p. 582)

Bahamut	
IPv4 Address	Reserved IP
10.0.0.3	
MAC Address	
78:24:AF:D7:3D:C9	
Comments	

```
Description . . . . . : Intel(R) Ethernet Connection (2) I218-V
Physical Address . . . . . : 78-24-AF-D7-3D-C9
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv4 Address. . . . . : 10.0.0.3(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Friday, September 11, 2020 8:28:43 AM
Lease Expires . . . . . : Friday, September 18, 2020 12:01:12 PM
Default Gateway . . . . . : 10.0.0.1
DHCP Server . . . . . : 10.0.0.1
```

13-24: Students will need to configure three devices with static IP addresses – the ESXi server, the management workstation (or a bastion host), and the WAN interface of the pfSense VM. There is no way around this. If for one reason or another it's not possible to set a static IP address for the management workstation, a possible work-around may be to configure a bastion host. Students will learn more about bastion hosts in chapter 16.

13.4.2: Connecting to the ESXi Web Interface

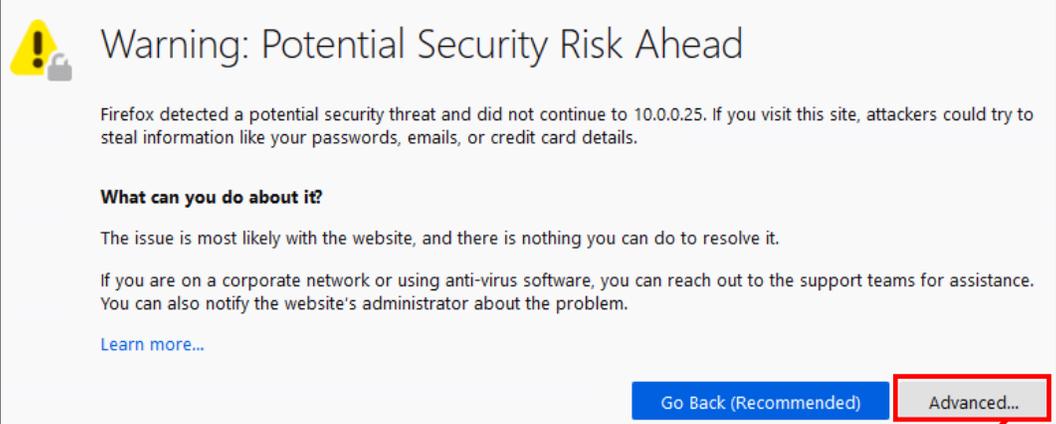
The ESXi web interface can be accessed from any client operating system, using any HTML5 compatible web browser. Please be aware that my personal web browser preference is Mozilla Firefox, and that while most of the illustrations in this chapter will be featuring Firefox, that Google Chrome (and all of its many variations – including Microsoft Edge) should be able to follow along easily.

Open your web browser, and enter the IP address of the ESXi server into the navigation bar like so:

```
https://10.0.0.25
```

With any luck, the web browser will present you with a screen similar to what is portrayed in *fig. 13-25* below. Most modern web browsers are extremely vocal about SSL certificates they don't trust. VMware ESXi uses a self-signed SSL certificate to serve the web interface over HTTPS. The web browser doesn't trust self-signed SSL certificates, and is throwing a fit over it. Fortunately, most web browsers have buttons or dialogue options along the lines of '*yes, I accept the risk, please just let me connect.*'

For example, Firefox presents the text, *Warning: Potential Security Risk Ahead*, trying to convince users that whatever site they are trying to access is probably up to no good. Next to the nicely highlighted button labeled *Go back (Recommended)* is a grey button labeled *Advanced* – click on it to open a window that explains why Firefox thinks the website is so bad. Turns out (as mentioned above), Firefox doesn't trust the certificate ESXi is using. Click the button labeled *Accept the Risk and Continue* to access the login screen. enter the username `root`, and the password configured for `root` from the ESXi installer, then click *Log in*.



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to 10.0.0.25. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

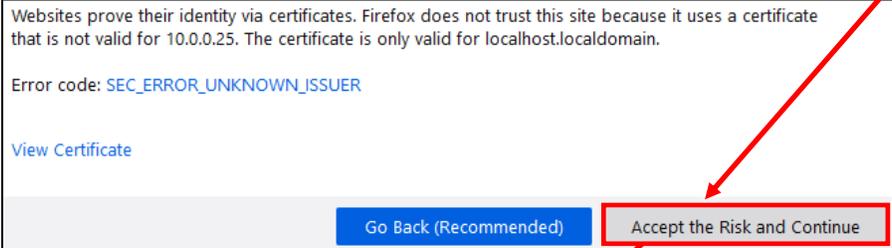
What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

[Learn more...](#)

[Go Back \(Recommended\)](#) [Advanced...](#)



Websites prove their identity via certificates. Firefox does not trust this site because it uses a certificate that is not valid for 10.0.0.25. The certificate is only valid for localhost.localdomain.

Error code: `SEC_ERROR_UNKNOWN_ISSUER`

[View Certificate](#)

[Go Back \(Recommended\)](#) [Accept the Risk and Continue](#)



User name:

Password:

[Log in](#)

vmware ESXi™

13-25: When students first access the ESXi web interface, most web browsers give users scary warnings of impending doom, but don't be alarmed. Click the *Advanced* button, then click *Accept the Risk and Continue*, or *Continue to [IP Address](unsafe)* to proceed. Enter the username `root`, and the password configured for the root user from the ESXi installer, then click *Log in*.

13.5: Configuring ESXi

There are three tasks for students to complete in this section before they can begin creating their lab virtual machines. **The first task will be to set up the free license key** students copied from vmware.com when they downloaded the ESXi ISO (refer back to [section 13.2.1](#), pp. 525-527). Currently, ESXi is running on a 60-day enterprise trial with a bunch of extra bells and whistles enabled. After those 60 days are up, the operating system requires a license key of some sort to continue operating.

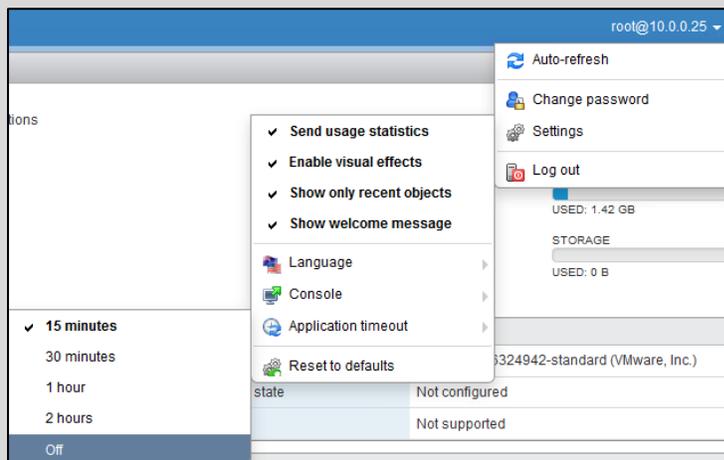
The next task will be creating the necessary virtual switches and port groups to support the network configuration of the lab environment. Recall back in chapter 4 learning about how virtual networking functions on most bare-metal hypervisors. ESXi uses virtual switches to provide network access, and virtual machines are assigned to port groups assigned to those virtual switches. Students will need to create four virtual switches and four corresponding port groups: Bridged, Virtual Management, IPS1, and IPS2.

The final task to perform will be to create a local datastore. A datastore is more or less VMware's terminology for a storage volume; it's where ESXi will store virtual machine files. While VMware lets you define remote datastores (e.g. NAS/Network Attached Storage, SAN/Storage Area Networking, iSCSI, etc.), **I'll be operating on the assumption that students will be using local storage** (e.g. hard disks and SSDs installed directly in the server itself). Students will create a Datastore, and upload the ISO files necessary for the lab environment. We'll also be covering how to import the Metasploitable 2 VM a little bit later.

Tired of Being Put in Time Out

You ever log into a web application, step away for a moment, then come back, only to find that the application you logged in has forgotten you existed and wants you to log back in? Happens to me a lot. By default, ESXi is configured to log off idle sessions after 15 minutes has passed. This can get really annoying, really quickly. Fortunately, there's an easy way to adjust this, or disable it entirely. Along the top of the ESXi web interface is the text [username]@[ip address] (e.g. root@10.0.0.25). If you click on this text, a menu appears that controls various settings for the currently logged in ESXi user. You can come here if you want to change your password, enable browser auto-refresh, and various other settings as needed.

Hover over the *Settings* option, then over the *Application timeout* settings. This setting controls how long before an idle session is logged out. My personal preference is to set it to *Off*, and lock my workstation when I need to step away for extended periods of time.

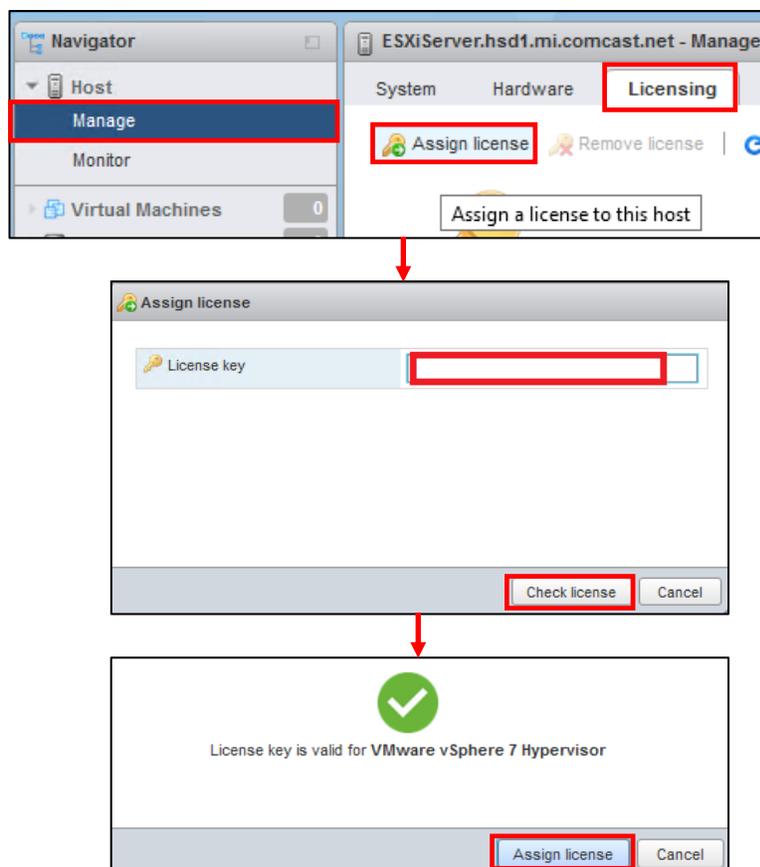


13-26: Someone thought that 15 minutes of inactivity before kicking off active sessions was a perfectly acceptable default. Turns out, this is a default setting for a few regulatory compliance standards. Fortunately, VMware makes it easy to adjust or altogether disable this feature.

13.5.1 Assigning a License

Upon first logging into the ESXi web interface, students are greeted the host information screen. This screen provides users with more information about their server. Along the left side of the browser window, is a small window pane labeled *Navigator*. We'll be using the navigator to perform other actions on the server momentarily. For now, take note that the *Host* option is highlighted, and that there are two options underneath it, labeled *Manage* and *Monitor*.

Click on the option under *Host* labeled *Manage*, and the primary window pane updates. On the *Manage* screen, there are multiple tabs that contain settings to determine how the ESXi server operates. Click the tab labeled *Licensing*. Immediately below licensing is a small key icon with a green button over it labeled *Assign license*. Students should click on the button, enter the license key they copied earlier, then click *Check License*. Once your license validates, click the *Assign license* button that appears. Students may notice that the number of supported features their license supplies is significantly less than the number of features the trial license provided. Unfortunately, with enterprise software, you get what you pay for. On the bright side however, students have everything they need to get their lab up and running.



13-27: To apply your free ESXi license, click the *Manage* option under *Host* on the *Navigator* pane, click on the *Licensing* tab, and click the *Assign license* button. A window pops up with an input box for users to enter their license key. Once entered, click the *Check license* button. If it's a valid license key, a green checkmark will appear, confirming that the key is valid. Click the *Assign license* button to finish setting the software license.

13.5.2 Virtual Switches and Port Groups

On the *Navigation* pane, click on *Networking*. Click on the tab labeled *Physical NICs*. On this page, ESXi will display all of the physical network interfaces it was able to detect, and whether or not they are physically connected to a network, via the *Link speed* column. The instructions provided in this section assume that students have at least two network interfaces on their ESXi server, and that they are both connected to their local, physical network (If students only have one network interface connected and available, see the sidebar discussion, *Close Quarters Networking*). If students haven't already connected their secondary network interface, connect it now, wait a few moments, and click the *Refresh* button (small blue circular arrow icon).

Next, click on the *Virtual switches* tab. By default, ESXi creates a virtual switch named *vswitch0*. This virtual switch is important. It contains details that allow us to connect to management interface, and thus the ESXi web interface. **Do not modify or delete vswitch0.** Do not taunt happy fun ball.

Click on the *Add standard virtual switch* button. Students will need to create four additional virtual switches with the follow settings:

Name:	Bridged	Virtual Management	IPS 1	IPS2
Uplink 1:	Secondary, connected interface	None! Click the grey, circular "X" and remove the Uplink 1 option.	None! Click the grey, circular "X" and remove the Uplink 1 option.	None! Click the grey, circular "X" and remove the Uplink 1 option.
Security Settings:				
Promiscuous Mode:	Reject	Reject	Accept	Accept
MAC address changes:	Reject	Reject	Accept	Accept
Forged transmits	Reject	Reject	Accept	Accept

For the Uplink 1 drop-down box, the only virtual switch that requires an uplink at all is the *Bridged* virtual switch. If students connected a secondary network interface as recommended, ESXi is usually smart enough to automatically detect that the interface is up, and assume that the user would like to use it as an uplink for the switch. If that automatic detection is incorrect, click the drop-down and select another network interface manually. **For the Virtual Management, IPS 1 and IPS 2 virtual switches, they do not require a network uplink and should not have an uplink network interface assigned.** To the right of the drop-down menu is a small, grey circular X icon. Click on it to remove the *Uplink 1* field entirely.

The security options can be accessed by clicking the *Security* field (the blue part, and ironically, not the part that says *Click to expand*). This causes the options *Promiscuous mode*, *MAC address changes*, and *Forged transmits* to appear. **By default, whenever a new virtual switch is created, all three settings are set to Reject.** So, for the Bridged and Virtual Management switches, there is no need to toggle the security settings. **For the IPS1 and IPS2 virtual switches however, all three settings must be set to Accept** in order for the IDS/IPS software, and AFPACKET bridging to work correctly.

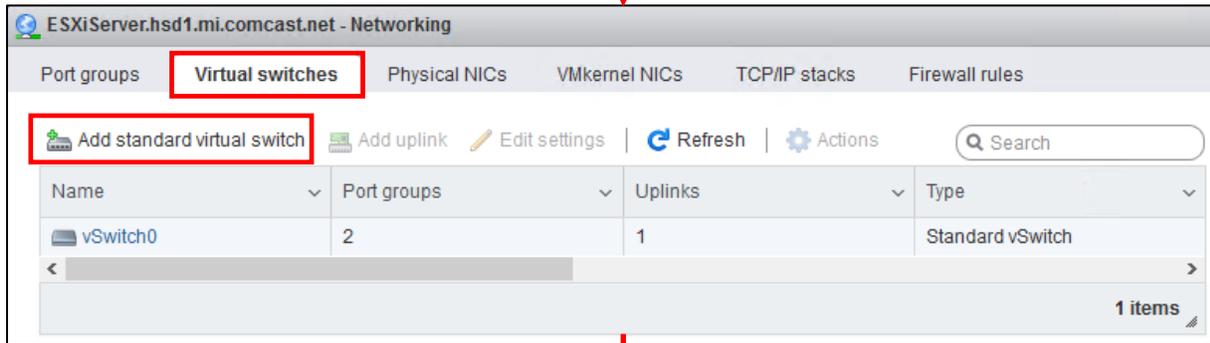
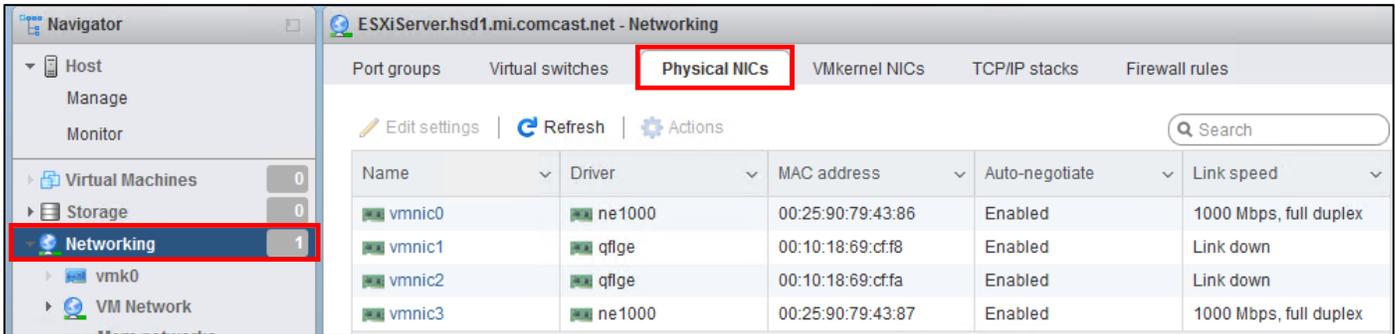
After creating the four virtual switches, click on the *Port groups* tab. In ESXi, VMs aren't directly attached to a virtual switch. Instead, they are assigned to a port group, and that port group is then assigned to a virtual switch. Users can define multiple port groups to a single virtual switch. This allows users to do things like set VLAN tags or unique security settings that only apply to a subset of virtual machines, instead of every single VM connected to the virtual switch.

By default, ESXi creates the *VM Network* and *Management Network* port groups, and assigns them both to *vSwitch0*. Right click on the *VM Network* port group, and click *Remove*. **Do not modify or remove the Management Network port group**, do not taunt happy fun hypervisor.

Next, students will need to create four port groups that correspond to the four virtual switches they recently created. Please use the following settings:

Port Group Name:	Bridged	Virtual Management	IPS1	IPS2
Virtual switch:	Bridged	Virtual Management	IPS1	IPS2

By default, the security settings for every port group a user creates should be set to *Inherit from vSwitch*. That means virtual machines assigned to a given port group will have their network security settings defined by the network security settings of the virtual switch that port group is associated to. For example, the *Bridged* port group, is assigned to the *Bridged* virtual switch. The security settings for that vSwitch were all set to the default of *Reject*. That means, virtual machines assigned to the *Bridged* port group will have *Promiscuous mode*, *MAC address changes*, and *Forged transmits* set to *Reject*.



Continue to *fig. 13-29*

13-28: The *Physical NICs* tab under *Networking* shows all of the physical network cards ESXi was able to identify. Notice *vmnic3* and *vmnic0* both have *1000Mbps, full duplex* for their *Link speed*, while the remaining interfaces have the status of *Link down*. From the previous sections, we know that *vmnic3* is the management interface for my ESXi server. That means *vmnic0* is the secondary network interface. Next up, click in the *Virtual switches* tab, then click *Add standard virtual switch*.

Continued from *fig. 13-28*

vSwitch Name	Bridged
MTU	1500
Uplink 1	vmnic0 - Up, 1000 mbps
▶ Link discovery	Click to expand
▼ Security	
Promiscuous mode	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
MAC address changes	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Forged transmits	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
<input type="button" value="Add"/>	

vSwitch Name	IPS1
MTU	1500
▶ Link discovery	Click to expand
▼ Security	
Promiscuous mode	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
MAC address changes	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Forged transmits	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
<input type="button" value="Add"/>	

vSwitch Name	IPS2
MTU	1500
▶ Link discovery	Click to expand
▼ Security	
Promiscuous mode	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
MAC address changes	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Forged transmits	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
<input type="button" value="Add"/>	

vSwitch Name	Virtual Management
MTU	1500
▶ Link discovery	Click to expand
▼ Security	
Promiscuous mode	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
MAC address changes	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Forged transmits	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
<input type="button" value="Add"/>	

Uplink 1	vmnic1 - Down	<input type="button" value="✕"/>
----------	---------------	----------------------------------

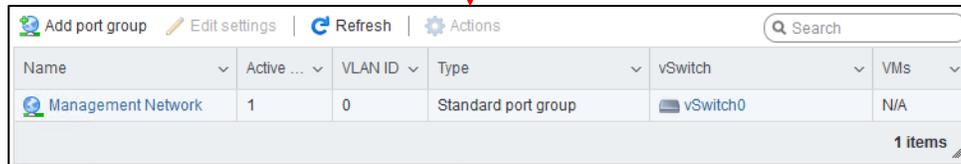
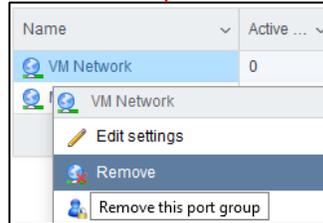
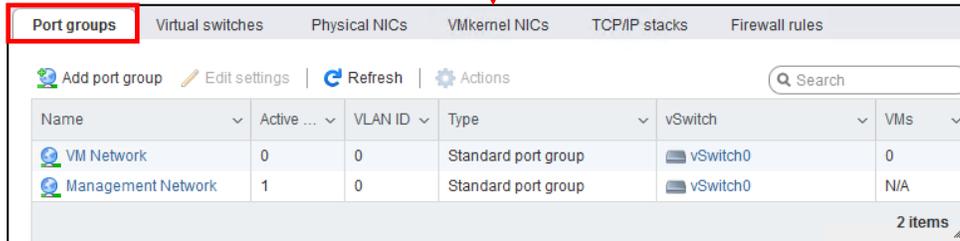
▶ Security	Click to expand
------------	-----------------

▼ Security	
Promiscuous mode	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
MAC address changes	<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Forged transmits	<input type="radio"/> Accept <input checked="" type="radio"/> Reject

Continue to *fig. 13-30*

13-29: Create four virtual switches with the settings displayed above. ***The only virtual switch that should have an uplink is the Bridged virtual switch.*** Click the light grey circle with an X in the middle on the far right of the *Uplink 1* field to remove the field entirely. Do this for the *Virtual Management*, *IPS1* and *IPS2* virtual switches. ***Be sure to set Promiscuous mode, MAC address changes, and Forged transmits to Accept for the IPS1 and IPS2 Virtual switches.***

Continued from *fig. 13-29*



Continue to *fig. 13-31*

13-30: Next, click on the *Port groups* tab. By default, ESXi creates two port groups *VM Network*, and *Management Network*. They are both assigned to *vSwitch0*. Right-click on *VM Network*, and click *Remove*. Next, click the *Add port group* button.

Continued from *fig. 13-30*

The figure displays four sequential screenshots of the 'Add port group' configuration window, each for a different mode. Each window has a title bar with a globe icon and the mode name. The fields are as follows:

- Add port group - Bridged:** Name: Bridged; VLAN ID: 0; Virtual switch: Bridged.
- Add port group - Virtual Management:** Name: Virtual Management; VLAN ID: 0; Virtual switch: Virtual Management.
- Add port group - IPS2:** Name: IPS2; VLAN ID: 0; Virtual switch: IPS2 (with a tooltip 'Virtual switch').
- Add port group - IPS1:** Name: IPS1; VLAN ID: 0; Virtual switch: IPS1.

In all four windows, the 'Security' section is collapsed and shows 'Click to expand'. An 'Add' button is located at the bottom right of each window.

Continue to *fig. 13-32*

13-31: Create four port groups with the settings displayed above. There should be no need to modify the security settings for any of these port groups.

Continued from *fig. 13-31*

Port groups						Virtual switches						Physical NICs						VMkernel NICs						TCP/IP stacks						Firewall rules					
Add standard virtual switch						Add uplink						Edit settings						Refresh						Actions						Search					
Name		Port groups		Uplinks		Type																													
vSwitch0		2		1		Standard vSwitch																													
Bridged		0		1		Standard vSwitch																													
Virtual Management		0		0		Standard vSwitch																													
IPS1		0		0		Standard vSwitch																													
IPS2		0		0		Standard vSwitch																													
5 items																																			

Port groups						Virtual switches						Physical NICs						VMkernel NICs						TCP/IP stacks						Firewall rules					
Add port group						Edit settings						Refresh						Actions						Search											
Name		Active ...		VLAN ID		Type		vSwitch		VMs																									
Management Network		1		0		Standard port group		vSwitch0		N/A																									
Bridged		0		0		Standard port group		Bridged		N/A																									
Virtual Management		0		0		Standard port group		Virtual Management		N/A																									
IPS1		0		0		Standard port group		IPS1		N/A																									
IPS2		0		0		Standard port group		IPS2		N/A																									
5 items																																			

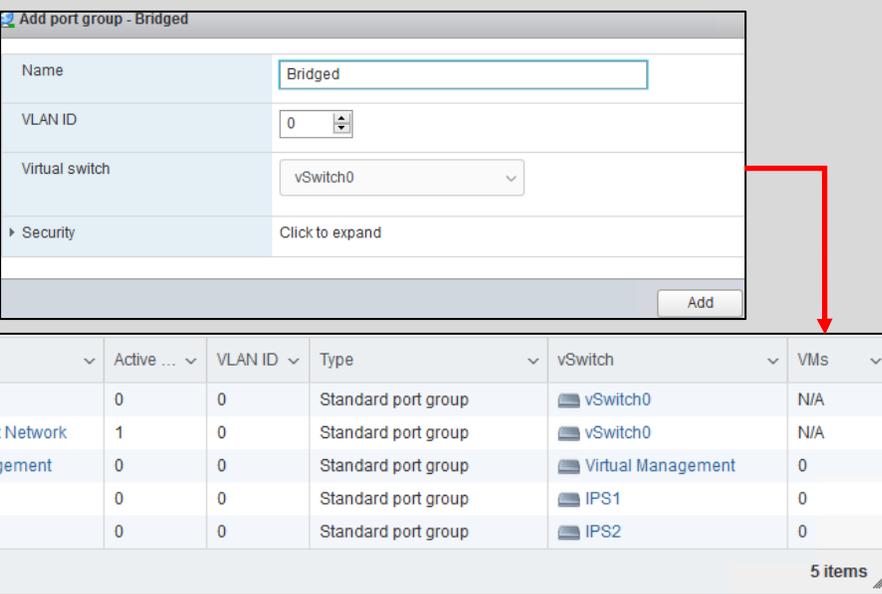
13-32: When students are finished, there should be five virtual switches in total on the *Virtual switches* tab, and five port groups in total on the *Port groups* tab.

Close Quarters Networking

Through some means, you have an ESXi server with only a single physical network interface available. You're probably wondering how to provide your lab environment with a virtual switch that has an uplink to a physical NIC. By default, ESXi creates *vSwitch0* and assigns the management network interface as the uplink. In my case, this was *vmnic3*. It also assigns the *Management Network*, and *VM Network* port groups to *vSwitch0*. I had you delete the *VM Network* port group to avoid any confusion when you have to create virtual machines and assign their interfaces to port groups later.

The work-around here is simple: Don't bother creating the *Bridged* virtual switch at all. If you already created a Bridged virtual switch without an uplink (or an inactive uplink) delete it by right clicking on it in the *Virtual switches* tab, then clicking the *Remove* option (just like we did with the VM network port group above). Note that if you create a *Bridged* port group and assigned it to this virtual switch, that port group will be deleted as well.

Next, navigate to the *Port groups* tab, and click the *Add port group* button. Name this new port group *Bridged*, and select *vSwitch0* from the *Virtual switch* drop-down menu. There you go, your lab now has an uplink to a physical network.



The screenshot shows the 'Add port group - Bridged' dialog box with the following fields:

- Name: Bridged
- VLAN ID: 0
- Virtual switch: vSwitch0
- Security: Click to expand

The 'Add' button is at the bottom right of the dialog. A red arrow points from this button to the 'Bridged' row in the table below.

Name	Active ...	VLAN ID	Type	vSwitch	VMs
Bridged	0	0	Standard port group	vSwitch0	N/A
Management Network	1	0	Standard port group	vSwitch0	N/A
Virtual Management	0	0	Standard port group	Virtual Management	0
IPS1	0	0	Standard port group	IPS1	0
IPS2	0	0	Standard port group	IPS2	0

5 items

13-33: If your ESXi server only has one physical NIC available, create the *Bridged* port group, and assign it to *vSwitch0* instead. You will have four virtual switches in total, and five port groups.

Please be aware that while this technically works, and is probably good enough for a lab environment, if the physical network interface becomes overloaded from too much network traffic, there is a possibility that you may lose connectivity to the ESXi web interface until the traffic subsides. This is the reason I recommended two network interfaces: To ensure management network traffic and virtual lab network traffic is carried by two, separate network interfaces.

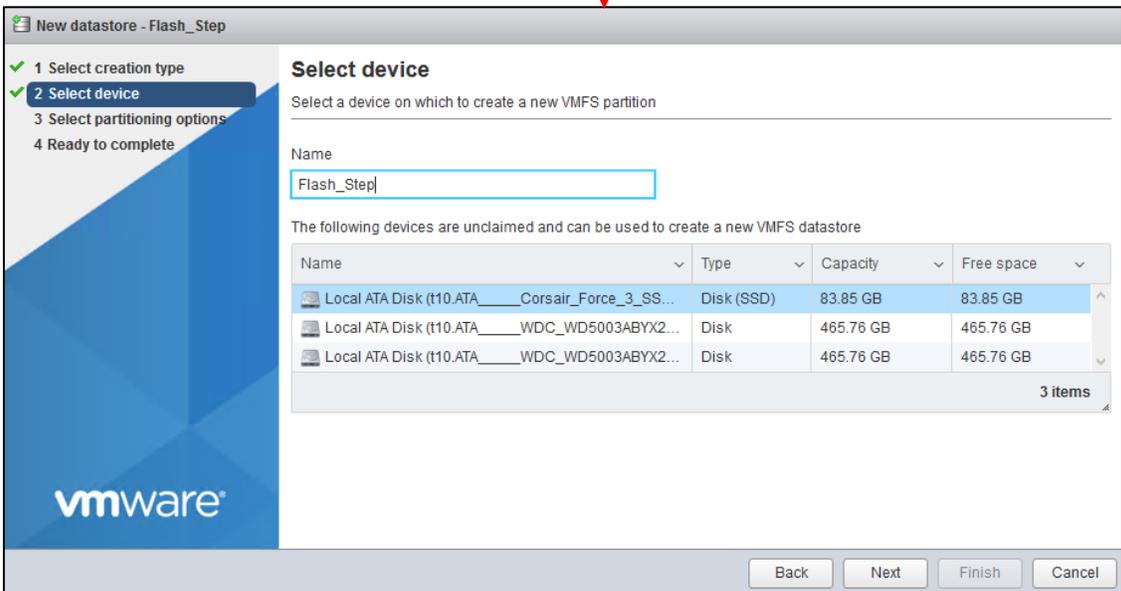
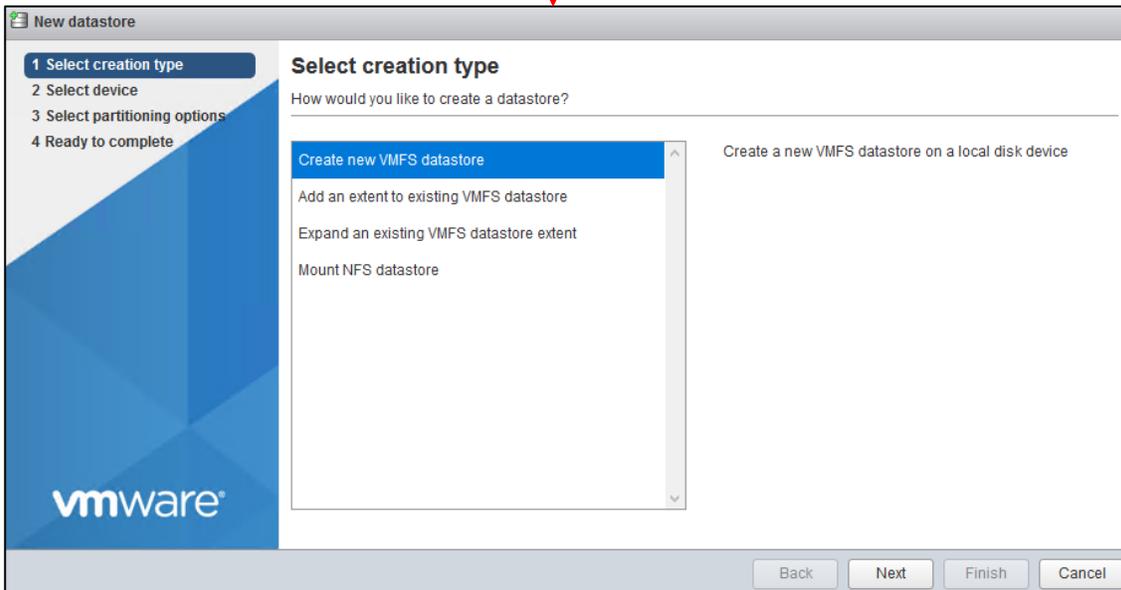
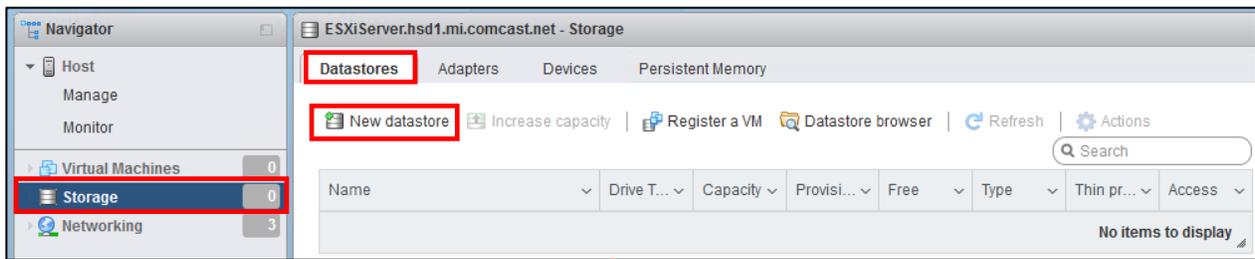
13.5.3: Datastores

As mentioned in section 13.5, Datastores are where ESXi holds all of the virtual machine files – This includes image ISOs as well. To get started, click on the *Storage* option under the Navigation pane. By default, the *Datastores* tab is displayed, and by default, ESXi will define no datastores. Click the *New datastore* option to begin creating your first datastore.

A window titled *New datastore* appears. This wizard will guide students through the process of creating a datastore. This first screen is labeled *Select creation type*. Click the option *Create new VMFS datastore* to highlight it, then click *Next* to continue. On the next screen, ESXi will show students a list of available storage drives that can be used to create a datastore. Please note that depending on how storage is configured on the physical server (e.g., number of attached drives, RAID arrays, etc.) students may have more or less options than what I am demonstrating on my server. Select a drive, enter a name for the datastore, then click *Next*. On the next screen, labeled *Select partitioning options*, ESXi lets you decide how you want to partition, or slice up the disk for storage. I recommend selecting the *Use full disk* option, and the default filesystem recommended by the wizard (in my case VMFS 6 was selected by default). Click *Next* to proceed. The final screen, labeled *Ready to complete*, confirms the user's choices they have made. Click the *Finish* button, and ESXi will display one final pop-up to warn users that if there was any data on the disk, it will be erased. Click *Yes* to proceed, and finish the new datastore wizard.

On the bottom of the page, there is a pane labeled *Recent tasks* that can be used to monitor the status of various tasks given to the ESXi server to perform. The most recent one should be the task *Create Vmfs Datastore*. Once the *Result* field has been updated to *Completed successfully*, click on the *Storage* option on the Navigation pane. This should refresh the *Datastores* tab, and cause the newly created datastore to appear. Students should run the New datastore wizard for every hard drive ESXi was able to detect. For example, ESXi found 3 drives on my system, so I will run the wizard two additional times in order to create two more datastores on those drives.

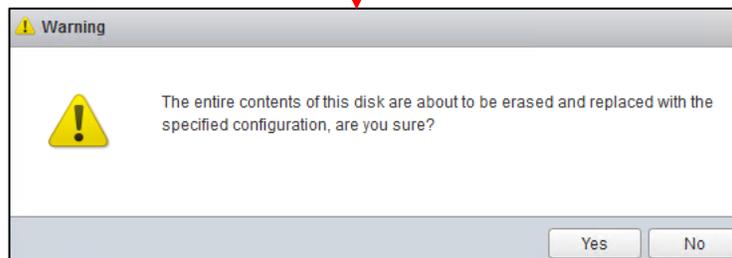
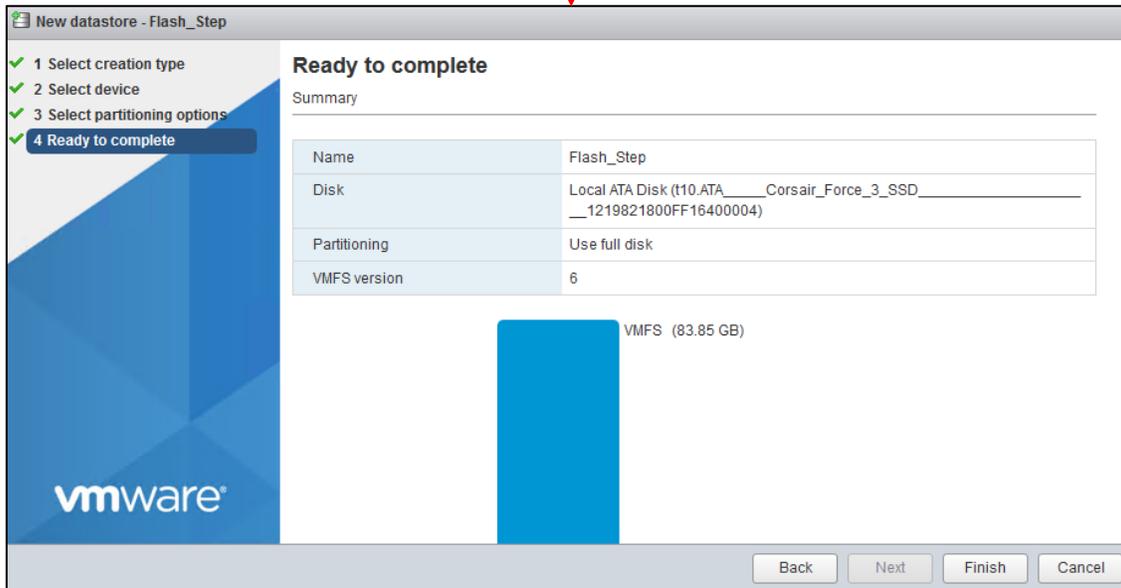
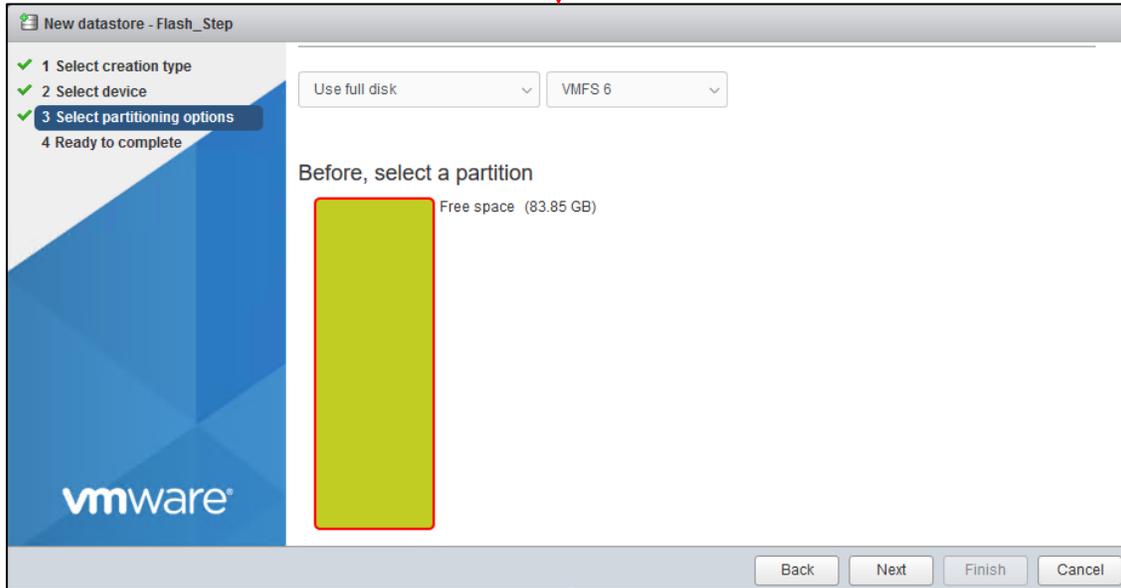
If students are seeking additional recommendations, or running into problems creating datastores on their ESXi server, see the sidebar conversations below *High-Level RAID*, and *Driven* for some troubleshooting recommendations.



Continued to *fig. 13-35*

13-34: Under the *Navigator* pane, Click on *Storage*. The *Datastores* tab should be selected by default. Click on the button labeled *New datastore* to begin the new datastore wizard. Select the *Create new VMFS datastore* option, then on the next screen, choose the drive to create the data store on.

Continued from *fig. 13-34*



Continued to *fig. 13-36*

13-35: For partitioning options. Select *Use full disk*, and the latest filesystem recommended by the datastore wizard. On the final screen, click *Finish*, followed by *Yes* on the warning pop-up that appears.

Continued from *fig. 13-35*

Task	Target	Initiator	Queued	Started	Result	Complete...
Create Vmfs Datastore	ESXiServer.hsd1.m...	root	09/17/2020 0...	09/17/2020 0...	Completed successfully	09/17/2020 0...

Name	Drive T...	Capacity	Provisi...	Free	Type	Thin p...	Access
Flash_Step	SSD	83.75 GB	1.41 GB	82.34 GB	VMFS6	Supported	Single

1 items

Two more runs of the new
datastore wizard later....

Name	Drive T...	Capacity	Provisi...	Free	Type	Thin p...	Access
Flash_Step	SSD	83.75 GB	1.41 GB	82.34 GB	VMFS6	Supported	Single
Masa	Non-SSD	465.5 GB	1.41 GB	464.09 GB	VMFS6	Supported	Single
Mune	Non-SSD	465.5 GB	1.41 GB	464.09 GB	VMFS6	Supported	Single

3 items

13-36: After a moment or two, refresh the *Datstores* tab, and the newly created datastore should appear. Repeat this process for any additional drives attached to the server.

High-Level RAID

It's very likely that the number of attached hard drives, solid state drives, or RAID arrays will be completely different from the number of drives I have installed on my server that I'm using for demonstration. Keeping this in mind, here are some recommendations for creating data stores and managing local storage on your server:

- **If you have a server with a physical RAID card/controller installed, absolutely take advantage of this and create a RAID array.** I recommend using RAID 1 (mirroring), RAID 5 (striping with parity) or RAID 10 (Striping + mirroring) depending on the number of disks you have installed in your server to support these RAID levels. If you aren't using a professional grade server, or have no idea what a RAID array is, don't worry about this for now.

If you do have this sort of hardware and have no idea what you're doing, I recommend looking up the manufacturer and model number for your server (e.g., Cisco UCS C240 M5, HP DL 360e, etc.). Pull up the documentation, and figure out how you're supposed to configure a RAID array.

If you don't know what a RAID array is, using your favorite web browser and search engine, look up the differences between RAID 1, RAID 5, and RAID 10, as well as how many hard drives you need for each of these RAID levels.

- **Remember that huge long conversation we had way back in section 13.1.2 about hardware compatibility? Your RAID controller (especially if it is a built-in "software" RAID controller) may not be compatible with ESXi.** Your drives may show up, but any RAID arrays you configure may or may not be recognized. Click on the *Devices* tab to see if ESXi was able to detect your hard drives. If they aren't there, unfortunately, there isn't much that can be done aside from acquiring a RAID controller that is compatible with ESXi.

- **If you are working with individual hard drives and/or SSDs (e.g., you have a bunch of attached disks, but they are not configured for RAID, or the RAID controller is not recognized), create a datastore for each hard drive and/or SSD (except the USB drive/SD card you installed ESXi to).** Yes, ESXi has the ability to dynamically expand data stores, and allocate more disks to a single datastore, but this is a really, really good way for you to suddenly lose half of your data when a drive fails, and not discover it until that one virtual machine you needed is dead. At least with separate datastores, you'll know immediately that your VMs are gone when you can't access the datastore anymore.

Driven

If you're having a problem where you know there's a drive available on your server, and the *Devices* tab confirms it, but the new datastore wizard doesn't want to let you create a datastore on it, there's probably already a partition or filesystem on it. here's the solution for that:

- Navigate to the *Devices* tab, and click on the drive that is giving you trouble.
- A new window opens in the center pane that provides information about the drive you clicked on.
- Click on the gear icon labeled *Actions*, then click on *Clear partition table*. A pop-up appears, warning that this will delete any data stored on the disk. Click the *Yes* button to assert your dominance over the machine legion, and delete the data.
- Finally, you can click on the *New datastore* button on the page and you'll get a slightly modified new datastore device wizard. Since you've already navigated to a specific drive, the wizard assumes you want to create a datastore on that particular drive. Finish the wizard and the drive should show up under the *Datastores* listing eventually.

The figure consists of four screenshots illustrating the process of clearing a partition table and creating a new datastore. The first screenshot shows the 'Devices' tab in a management interface, listing several disks. The disk 'Local ATA Disk (t10.ATA...ST1000DM003...)' is highlighted with a red box. The second screenshot shows the details page for this disk, with the 'Actions' menu open and 'Clear partition table' highlighted in red. The third screenshot shows a warning dialog box titled 'Confirm clear partition table' with the 'Yes' button highlighted in red. The fourth screenshot shows the details page for the disk with the 'New datastore' button highlighted in red.

13-37: If the *Devices* tab shows a drive, but the new datastore wizard doesn't see it, try clearing its partition table, then clicking the *New datastore* option. To start a slightly modified version of the new datastore wizard.

13.5.3.1: Staging

Now that students have at least one datastore for their virtual machines, the next step is to upload the ISO files we will be using to install operating systems to those VMs. Students should refer to [section 1.5.4](#) (p. 26) for a list of ISOs and virtual machine images needed for their lab environment. To begin, if students aren't there already, click on *Storage* under the *Navigator* pane, select the *Datastores* tab if it isn't already selected, then click the icon with a spyglass and a folder labeled *Datastore browser*.

This opens a window labeled datastore browser that has three panes. The first pane is a list of datastores that the ESXi server is aware of. If students have more than one datastore on their server, they may select another datastore to use for storing their ISO files. I recommend creating a folder on the datastore, labeled ISOs just to make it easier to find them later on while creating virtual machines later. To do, click the folder with a small green plus sign labeled *Create directory*. A small pop-up box appears with an input box labeled *Directory name*. Under the input box, ESXi will tell you where the folder is going to be created. For example, on my server, I'm making the ISOs directory in the root folder of the [Flash_Step] datastore. The full path is:

[Flash_Step]/ISOs/

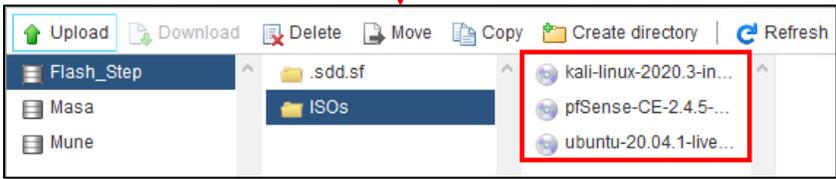
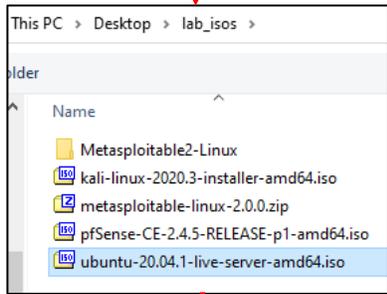
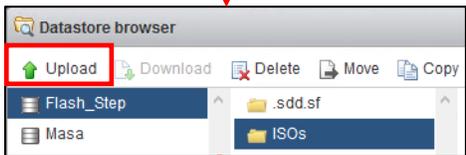
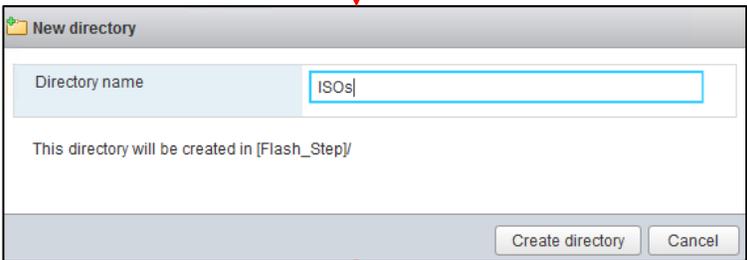
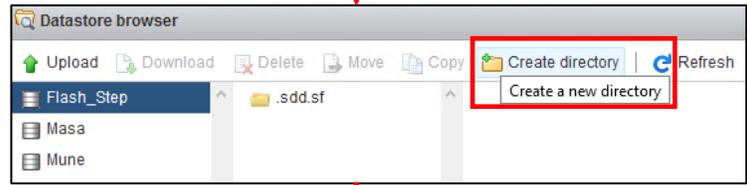
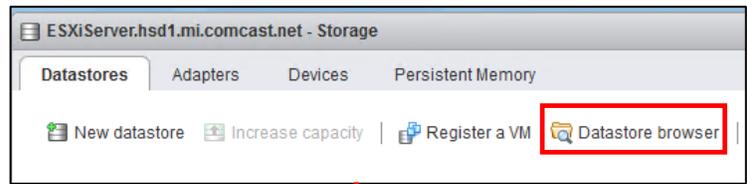
Name your directory and click the *Create directory* button. Next, click on the folder in the second pane to highlight it, then click the Green, upward facing arrow labeled *Upload*. This should open a file browser on your local computer that you can use to browse and select the ISO file you wish to upload. **Students will need to upload at least 3 ISOs in total:**

- The latest Kali Linux ISO

- The latest Ubuntu Server LTS ISO

- The latest decompressed pfSense ISO. Refer back to chapter 1, [section 1.8, Using Compression Tools](#) (pp. 33-35) for guidance on how to decompress the pfSense ISO.

When students are finished, all three iso files should be displayed on the third pane of the Datastore browser window, under the directory you created. **Don't worry about the Metasploitable 2 VM/zip file for now. Importing that takes a little bit more work that will be covered later in this chapter.**



13-38: Click on *Storage* in the *Navigator* tab, click the *Datastores* tab (if you aren't already there), then click the *Datastore browser* option. In the datastore browser, select a datastore in the far-left pane, then click the *Create directory* option. I recommend naming the directory *ISOs* for simplicity's sake. Next, highlight the new folder, click the *Upload* button and browse to the Kali Linux, pfSense, and Ubuntu Server 20.04 ISOs and upload each of them. Don't worry about the Metasploitable 2 VM, we'll be handling that later.

13.6 Building the first Virtual Machine, pfSense

The pfSense virtual machine is responsible for binding the entire lab environment together. It is a well-supported firewall distribution with amazing ease of use and functionality. pfSense is also very modular, featuring a system for adding on additional functionality through BSD's pkg software package manager.

Please ensure that the decompressed pfSense, Kali, and Ubuntu Server ISOs have been uploaded to the ESXi server's datastore before continuing. This process was covered in section 13.5.3.1.

13.6.1 VM Creation

Begin by clicking on the *Virtual Machines* option under the *Navigator* pane. Click the icon with a small green plus sign and overlapping blue squares labeled *Create / Register VM* to start the new virtual machine wizard. The first screen, labeled *Select creation type*, asks users how they would like to create a Virtual Machine. Click the option *Create a new virtual machine* to highlight it, then click *Next*.

The next screen is labeled *Select a name and guest OS*. In the *Name* input box, enter pfSense. For the *Guest OS family* drop-down, select *Other*, and for the *Guest OS Version*, select *FreeBSD 11 (64-bit)*. Click *Next* to proceed.

Note: pfSense CE is based on FreeBSD. If you're reading this in the future, Netgate (the makers of pfSense) has a support page that details what version of FreeBSD each version of pfSense is based on here:

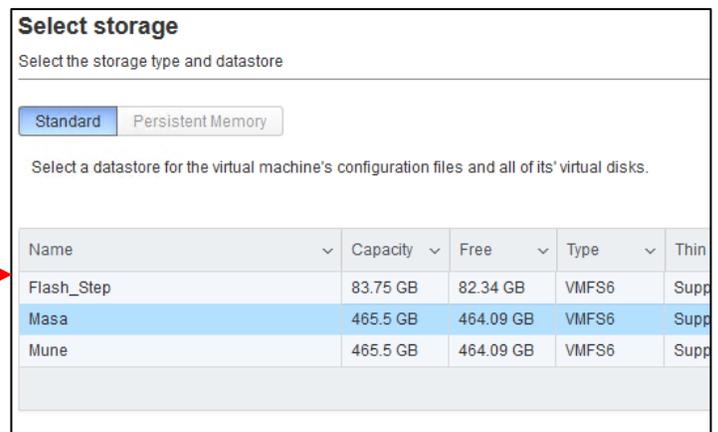
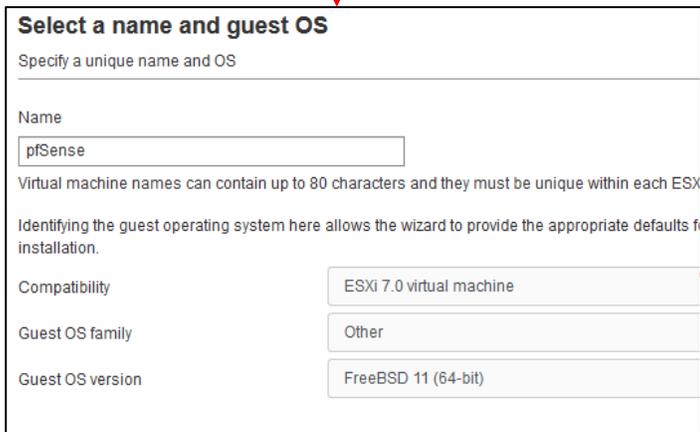
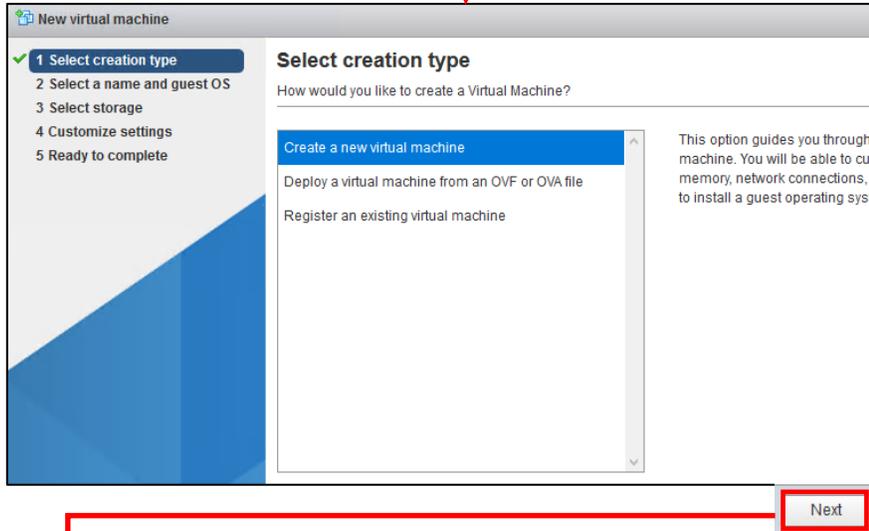
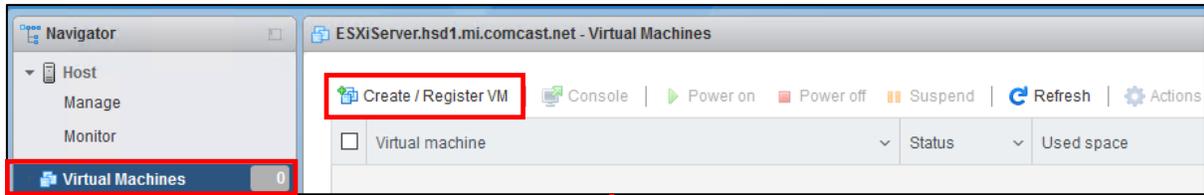
<https://docs.netgate.com/pfsense/en/latest/releases/versions-of-pfsense-and-freebsd.html>

The *Select storage* screen is next. Click on a datastore from the list available to store the VM, then click *Next*. That brings students to the *Customize settings* screen, and a slew of configuration settings to change. Perform the following configuration changes:

- Find the small, green button that looks like a network card labeled *Add network adapter*, and click it two times. **The pfSense VM needs three virtual network interfaces in total.**
 - Ensure the *Connect* checkbox is checked for all three network cards and modify the port groups drop-down each network card:
 - Network Adapter 1: Bridged
 - New Network Adapter (1): Virtual Management
 - New Network Adapter (2): IPS1
- Change the input box in the *Memory* section from the default of 1024MB, down to 512MB
- In the *Hard disk 1* input box, change the disk size from 8GB down to 5GB
 - Click the blue *Hard disk 1* box itself. This reveals a bunch of other hard disk settings as well. In the box labeled *Disk Provisioning*, click the *Thick provisioned, eagerly zeroed* radio button.
- Scroll down to the box labeled *USB controller 1*. Click the grey circle with a white X in the middle to the far right of the USB 2.0 drop-down box. This removes the USB controller from the virtual machine.
- Find the box labeled *CD/DVD Drive 1*. Ensure that the *Connect* checkbox is checked, then click the drop-down menu that has the value *Host device*, and change that to *Datastore ISO file*. This opens a window labeled *Datastore browser*. Navigate to the datastore and folder created in section 13.5.3.1 that contains all of the ISOs for the lab environment. Click the pfSense ISO to highlight it, then click the *Select* button to close the window.

After performing all of these configuration changes, click the *Next* button to continue. The final screen, labeled *Ready to complete*, contains a summary of all the configuration options that will be performed to create the new virtual machine. After reviewing the settings, click *Finish* to complete the new virtual machine wizard. The task *Create VM* will appear in the *Recent tasks* pane. Once the *Result* column changes to *Completed successfully*, the VM should appear in the table list on the *Virtual Machines* page.

Note: The *Thick provision, eagerly zeroed* option makes virtual machine creation take quite a bit longer to complete compared to the other disk provisioning options available. The reason I'm making you choose it is to force the hypervisor to allocate all of the disk space requested for the virtual machine immediately, and zero out that allocated disk space. There are other, faster options, but they all have drawbacks that can impact disk performance later. We wanna go fast.



Continue to *fig. 13-40*

13-39: Select the *Virtual Machines* option under the *Navigator* pane, then click the *Create/Register VM* button to start the new virtual machine wizard. On the first screen, select the *Create a new virtual machine* option. On the second screen, name the virtual machine *pfSense*, then set the *Guest OS family* to *Other*, and *Guest OS version* to *FreeBSD 11 (64-bit)*. Students should not have to modify the *Compatibility* drop-down option. On the third screen, choose a datastore to hold the virtual machine's files.

Continued from *fig. 13-39*

Customize settings

Configure the virtual machine hardware and virtual machine additional options

Add hard disk Add network adapter Add other device

Click here to display options related to *Hard disk 1*.

Students need to modify the *Disk Provisioning* setting.

Click this button twice to add two additional network cards. ***pfSense needs 3 NICs in total. Attach them to the port groups listed below.***

Click this icon to remove *USB controller 1*.

In the *CD/DVD Drive 1* setting, change the drop-down to *Datastore ISO file*.

This loads the datastore browser. Locate the *pfSense* ISO uploaded in the previous section, and select it.

Select

Cancel

Next

Continue to *fig. 13-41*

13-40: Welcome to the *Customize settings* screen. There is a lot that needs to be configured. Please be aware that only the configuration options that students need to change are displayed above. The notes in the margins above will help show you how to access and change some of the harder to find configuration settings. Once students are finished copying the configuration changes above, Click the *Next* button to proceed to the final screen.

Continued from *fig. 13-40*

Ready to complete

Review your settings selection before finishing the wizard

Name	pfSense
Datastore	Masa
Guest OS name	FreeBSD 11 (64-bit)
Compatibility	ESXi 7.0 virtual machine
vCPUs	1
Memory	512 MB
Network adapters	3
Network adapter 1 network	Bridged
Network adapter 1 type	VMXNET 3
Network adapter 2 network	Virtual Management
Network adapter 2 type	VMXNET 3
Network adapter 3 network	IPS1
Network adapter 3 type	VMXNET 3
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SCSI controller 0	LSI Logic SAS
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	5GB
Datastore	[Masa] pfSense/
Mode	Dependent
Provisioning	Thick provisioned, eagerly zeroed
Controller	SCSI controller 0 : 0
CD/DVD drive 1	
Backing	[Flash_Step] ISOs/pfSense-CE-2.4.5-RELEASE-p1-amd64.iso
Connected	Yes

Back Next **Finish**

ESXiServer.hsd1.mi.comcast.net - Virtual Machines

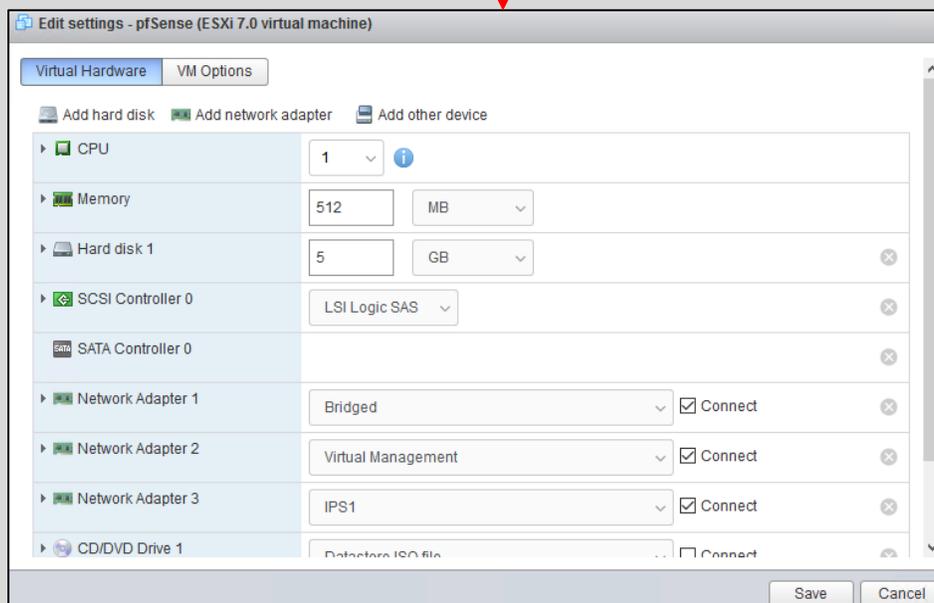
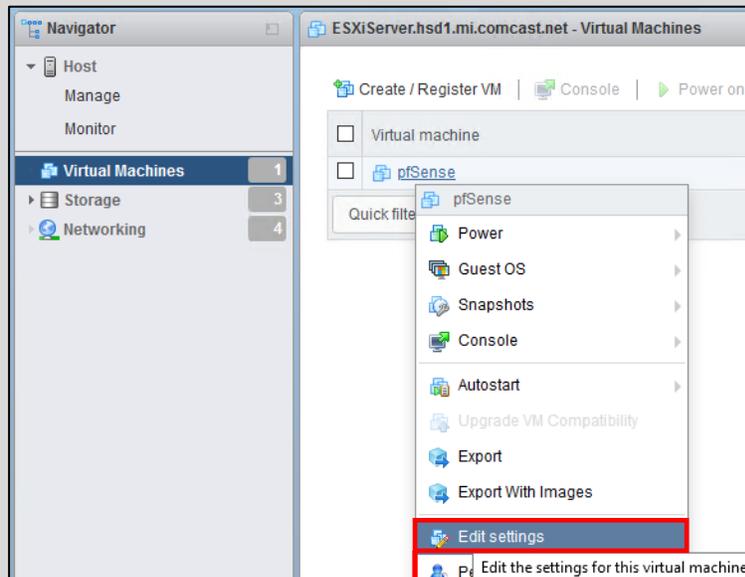
Create / Register VM | Console | Power on | Power off | Suspend | Refresh

<input type="checkbox"/>	Virtual machine	Status
<input type="checkbox"/>	pfSense	Normal

13-41: The final screen of the new virtual machine wizard has students review all the parameters they selected for creating the first virtual machine. Students should ensure that their *Ready to complete* screen closely matches the illustration above. The most important settings are the three virtual network adapters (and their associated port groups), removing the USB controller, as well as ensuring *CD/DVD drive 1* is configured to use the pfSense ISO file previously uploaded. Click the *Finish* button to complete the wizard. After a moment or two, the pfSense VM will appear on the *Virtual Machines* page.

Imperfect

If you created the pfSense virtual machine then suddenly realized there is a configuration setting you forgot to edit, don't fret. Under the *Navigator* pane, Click on *Virtual Machines* again to bring the virtual machine listing up, if you aren't already there. Right click on the pfSense VM, and select the icon with a small yellow icon and the overlapping blue squares labeled *Edit settings*. This causes a screen to appear that is almost identical to the Customize settings screen of the new virtual machine wizard – including the ability to add and remove hardware. Note that some hardware and/or configurations cannot be done while the virtual machine is powered on.



13-42: Missed a setting while creating your virtual machine? Navigate to the *Virtual Machines* listing, right click on the virtual machine in the table, and select *Edit settings*. A window appears that is nearly identical to the Customize settings screen of the new virtual machine wizard. Some changes cannot be done unless the VM is in the powered off state. Click the *Save* button to confirm any changes you make to your virtual machine.

13.6.2 First Boot and OS Installation

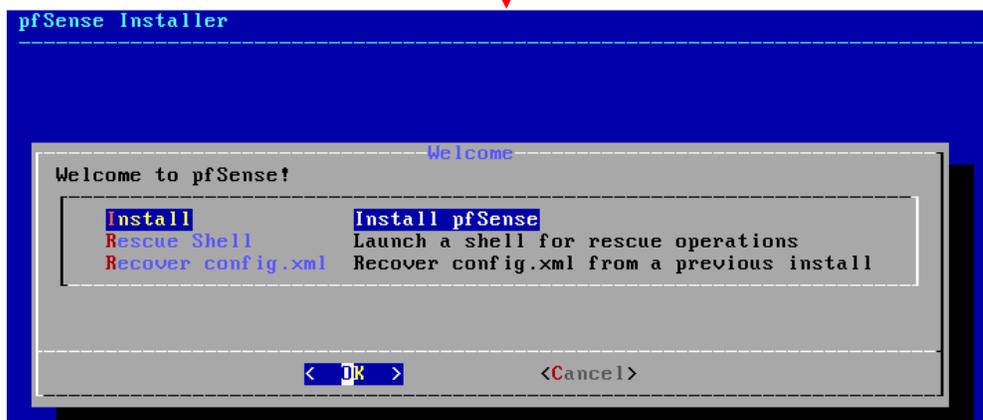
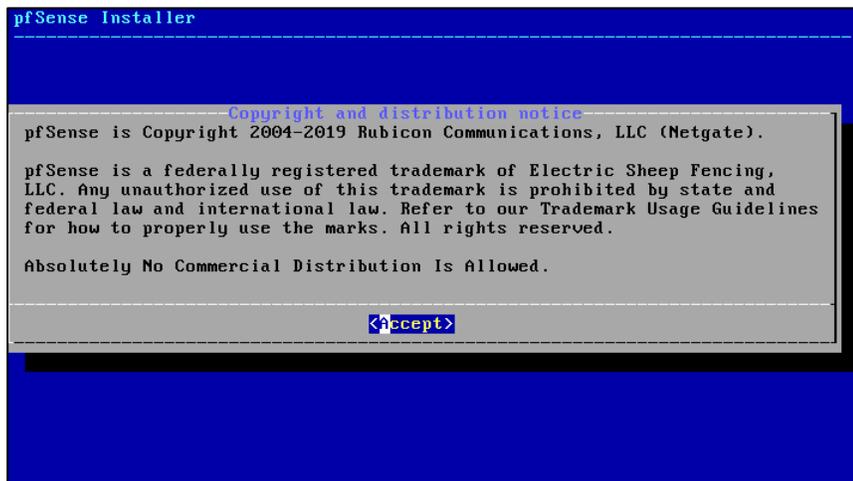
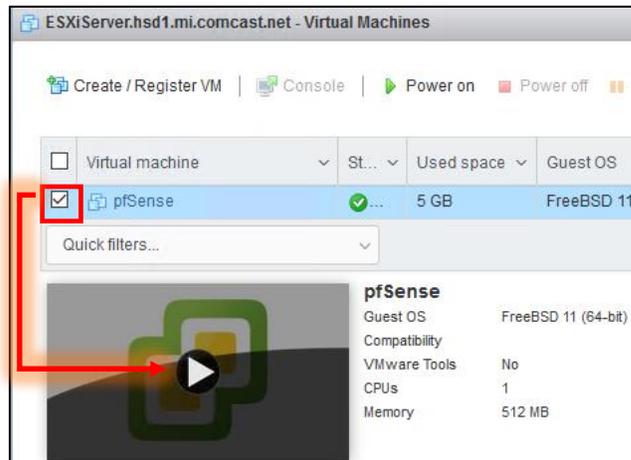
Our virtual machine has been created, and it is now time to install the pfSense operating system to the new VM. To begin, students will need to power on the virtual machine, then connect to its virtual console. There are multiple ways to do this, but the easiest method is to click the checkbox next to pfSense on the *Virtual Machines* pane, and in the window that appears beneath the virtual machines listing, click the icon that looks like a play button. This will both start the selected virtual machine and connect to the virtual console.

Think of this window as a direct keyboard, video, and mouse connection to the virtual machine while it is running. You'll notice a lot of text flying by as the VM boots from the installation ISO. Eventually you will reach the pfSense Installer. The first screen shows the *Copyright and distribution notice* for the software. Click anywhere in the virtual console window, and hit Enter to accept the software terms and conditions.

Next is the *Welcome* screen for the OS Installer. The option *Install pfSense* should be highlighted by default, but if not, use the arrow keys on your keyboard to select it, then hit enter. The next screen, titled *Keymap Selection* appears. If students are from a region of the world with a unique keyboard layout, they will need to search for and select it. Otherwise, select *Continue with default keymap* to use the US keymap, and hit enter. Next is the *Partitioning* screen. Partitioning is used to tell the installer how much and what portion of the disk to allocate. Since this is a virtual machine, and the disk is relatively tiny (5GB), select *Auto (UFS) Guided Desk Setup* and press enter to tell the installer to use all of the available disk space.

The installer handles formatting the disk and copying the operating system files over. The next screen, titled *Manual Configuration* asks if you want a command shell to manually edit any operating system files before closing the installer. Select *No*, and hit enter again. Finally, on the *Complete* screen, select *Reboot*, and hit enter. Congrats! You just installed the pfSense firewall distribution to your pfSense virtual machine.

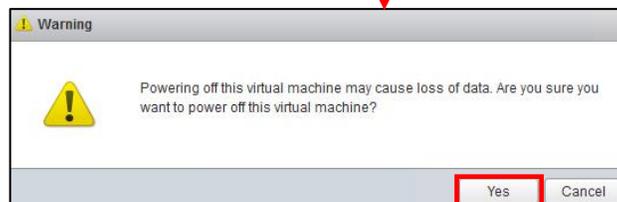
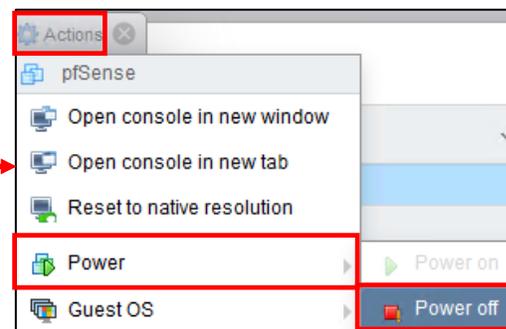
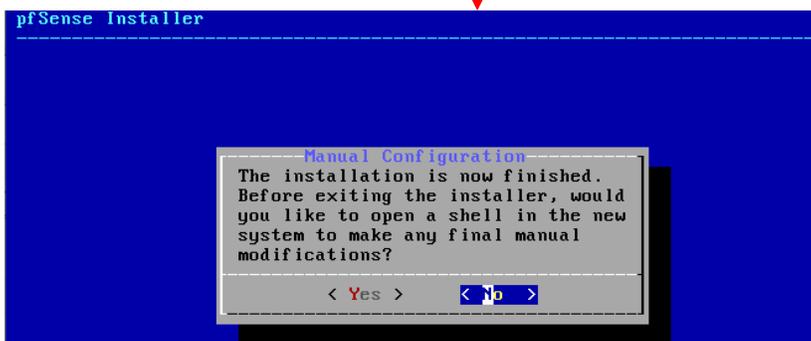
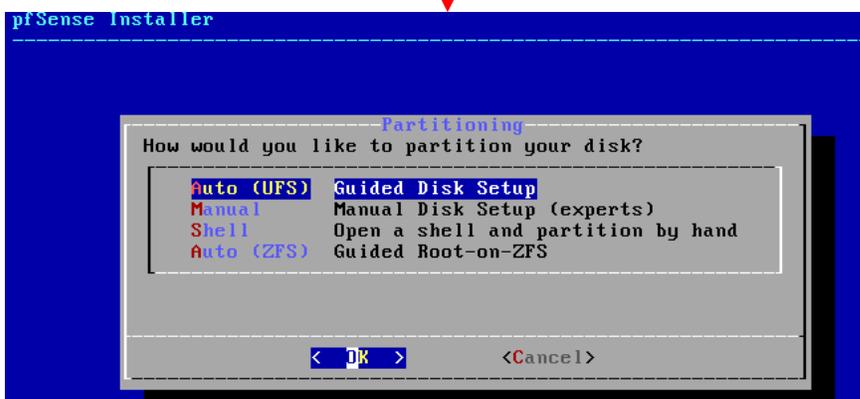
While the system is rebooting, click the icon with a gear in the upper right corner of the virtual console labeled *Actions*. A menu will appear. Select the *Power* menu option, then *Power off* from the sub-menu.



Continue to *fig. 13-44*

13-43: Power on the newly created pfSense VM, and connect to the virtual console. The VM should boot from the installation ISO automatically. Accept the License Agreement, then select the *Install pfSense* option to proceed.

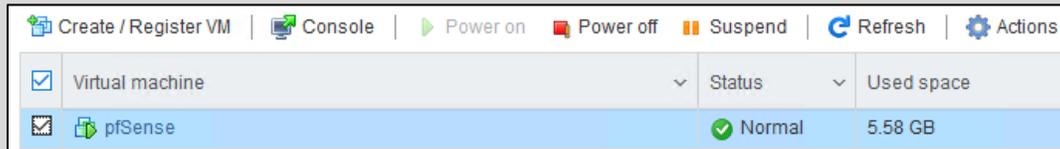
Continued from *fig. 13-43*



13-44: The pfSense installation process is pretty straightforward. Most students will be able to hit enter the entire way through and accept the defaults. Upon reaching the *Complete* screen, power off the virtual machine.

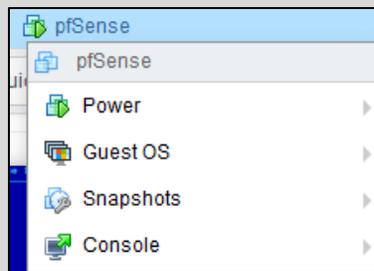
Changer of Ways

There are numerous places where users can power on, power off and/or connect to the virtual console of their virtual machines on the ESXi web interface. Clicking the checkbox next to a virtual machine's name enables the buttons immediately above the list of virtual machines (e.g., *Console, Power on, Power off*, etc.).



13-45: Click the checkbox next to a virtual machine's name enables the action buttons above the table list of virtual machines. What's more is that this can be used to power on/power off, and/or connect to the console of multiple virtual machines at once.

Alternatively, right-clicking on a virtual machine's name opens a context menu with both Console and Power options.



13-46: Right-clicking on a virtual machine's name opens a context menu that looks identical to the actions menu you accessed from the pfSense virtual machine's console.

In most cases, the virtual machine must be powered on before you can connect to its console.

13.6.3 pfSense Virtual Machine Settings

With the pfSense virtual machine powered off, access its settings menu. On the *Virtual Machines* page, right-click on the pfSense entry and select *Edit settings*. This brings up a window titled Edit settings. It's nearly identical to the Customize settings screen on the new virtual machine wizard. In fact, students who read the ~~im~~perfect side bar discussion already know how to get here.

There are a few things students need to do here before we can continue:

- Since the pfSense operating system is installed, the VM no longer needs the virtual CD/DVD drive. Scroll down to *CD/DVD Drive 1*, click the small grey circle with a white X in the middle to remove it.
- *CD/DVD Drive 1* was the only hardware that was attached to the SATA controller. With the drive removed, we no longer need that hardware, either. Scroll up to *SATA Controller 0* and click the same grey circle with a white X to remove that hardware as well.
- Students will need to record the MAC address for Network Adapters 1, 2 and 3. To do this, click on the name of the network adapter (e.g., *Network Adapter 1*) and this will cause additional fields to appear. Record the contents in the input box for the *MAC Address* field. **Note the network adapter number, MAC address, and the port group to which it is assigned for all three network interfaces.**

When finished, click the *Save* button to confirm these changes, and exit the *Edit settings* menu.

MAC and Cheese

You're probably wondering: *Why didn't we just record the MAC addresses of all three network interfaces on the Customize settings screen of the new virtual machine wizard?* It's mainly due to laziness on the part of the hypervisor. See, when ESXi first creates a network adapter, it doesn't assign it a MAC address until the first time the virtual machine is booted. This means that the MAC addresses you recorded in the section above were not present until after we started the pfSense virtual machine in order to install its operating system. This is will be somewhat important later for the remaining virtual machines in our lab environment, so just keep this in mind as we proceed.

What CD/DVD Drive?

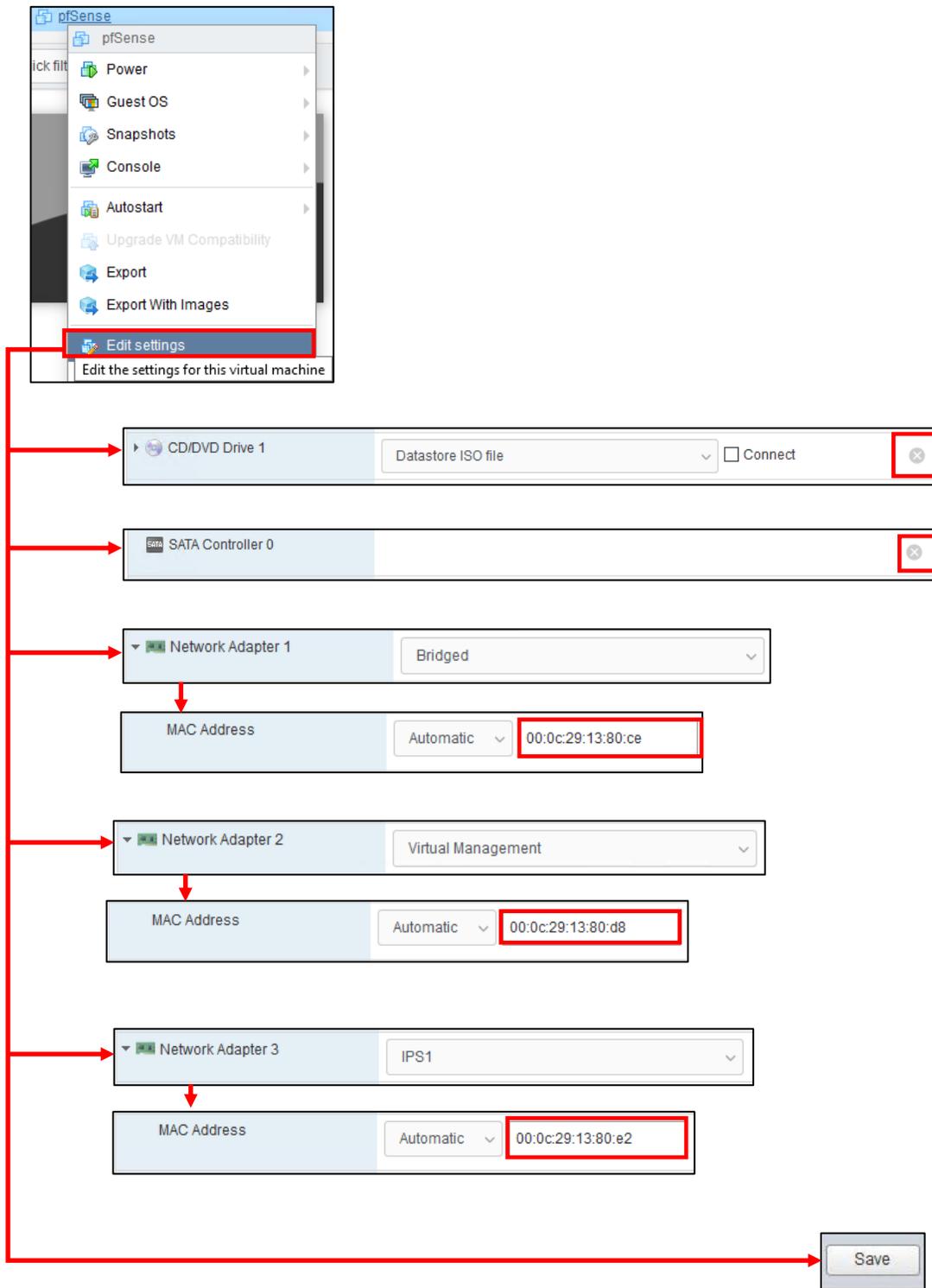
You may get the following error when attempting to remove *SATA Controller 0*, even after removing the *CD/DVD Drive 1* virtual hardware:



A CD/DVD drive is still attached to this controller, please reassign it first.

13-47: LIAR. I just removed it! Save your changes, and re-open the *Edit settings* menu if this happens to you.

If you get this error when attempting to remove the SATA controller, click *Save* to exit the *Edit settings* menu, then re-enter the menu by right clicking on the pfSense listing again. This time you should be able to successfully remove the *SATA Controller 0* virtual hardware.



13-48: Access the Edit settings menu by right clicking on the target virtual machine (on the Virtual Machines pane), and click the *Edit settings* option. ***Remove CD/DVD Drive 1, followed by SATA Controller 0 – in that specific order.*** This can be done for both devices by clicking the grey circle with a white X. Next up, click on *Network Adapter 1*, *Network Adapter 2*, and *Network adapter 3*. This causes additional fields to appear beneath all three network interfaces. ***Record the contents of the MAC Address input box for all three interfaces. Note the network adapter number and port group for each MAC address.*** Click the *Save* button to confirm these changes.

Noting the Notable

I can't overstate the value of documenting your lab network properly. Use whatever note-taking methods you prefer – paper and pen, Evernote, text editors, personal wikis, databases, spreadsheets, etc. Document the name of the VM, Operating system, the number of CPU cores allocated, RAM, Disk size, number of network adapters, network segments they are attached to, and their MAC addresses. This is called *asset management*, and it's an important habit to cultivate. Here is a template you can use for documenting your VMs:

VM Name:
Operating System:
CPU Cores:
RAM:
Disk Size:
Virtual Network Adapters:
Network Adapter #:
-Network Segment:
-MAC Address:
<Repeat for each network adapter>
Additional Notes:

And as an example, here is my pfSense VM entry:

VM Name: pfSense
Operating System: pfSense (FreeBSD)
CPU Cores: 1
RAM: 512MB
Disk Size: 5GB
Virtual Network Adapters: 3
Network Adapter 1:
-Network Segment: Bridged Port Group/WAN
-MAC Address: 00:0C:29:13:80:CE
Network Adapter 2:
-Network Segment: Virtual Management Port Group/LAN
-MAC Address: 00:0C:29:13:80:D8
Network Adapter 3:
-Network Segment: IPS 1 Port Group/OPT1
MAC Address: 00:0C:29:13:80:E2
Additional Notes: Lab firewall. Provides NTP, DNS, DHCP,
and HTTP proxy services.

Do this for every single virtual machine you add to your lab environment. Keep track of systems added or removed from the lab network. Always be aware of what's running on your networks. If you can do these things, you'll be better at asset management than most of the Fortune 500.

13.6.3.1 Static IP Address/DHCP Reservation for the Bridged/WAN MAC Address

In section 13.4.1, students learned the importance providing the Management interface for the ESXi server a static IP address. In the sidebar discussion in that section, *Static Cling*, it was also discussed that in total, students will need to provide at least three static IP addresses for their lab environment – either via static DHCP allocations, or through manual IP address configuration:

- The management interface of the ESXi server
- The management workstation, or a bastion host/jump box
- The network interface connected to the *Bridged* port group on the pfSense VM.

At this point, students should create a static DHCP mapping for the interface attached to the *Bridged* port group on the pfSense Virtual Machine. Again, every piece of network equipment has a different method for doing this and/or a different name for this functionality, but for reference, *fig. 13-49* below is another screen cap from the router my ISP provided, adding a static DHCP mapping for the pfSense VM.



The screenshot shows a web form titled "Add Device with Reserved IP Address". It contains the following fields:

- Host Name: pfSenseESXi
- MAC Address: 00:0C:29:13:80:CE
- Reserved IP Address: 10.0.0.26
- Comments: WAN interface of pfSense VM on ESXi Server.

At the bottom of the form are two buttons: "SAVE" and "CANCEL".

13-49: Create another static DHCP mapping for the interface on the pfSense VM attached to the *Bridged* port group. **This is extremely important in order to maintain access to the pfSense web interface later in this chapter (and in chapter 14).** If students operate on a network without DHCP services available, we'll go over the process of setting a static IP address shortly.

13.6.4 pfSense Command-Line and initial interface configuration

In this section, readers will navigate the command-line interface of their pfSense virtual machine to perform essential setup tasks. Once completed, users can navigate to the webConfigurator interface. Start the pfSense VM, then connect to its virtual console (refer to section 13.6.2, if students need a refresher).

13.6.4.1 The Assign Interfaces Wizard

After a few moments, the boot process completes and students are greeted by the *Assign Interfaces* wizard. This wizard is used to map our virtual machine's network interfaces (*Adapter 1*, *Adapter 2*, and *Adapter 3*) to their pfSense aliases – *WAN*, *LAN*, or *OPT1*. Unfortunately, the operating system itself also has unique names for each of these interfaces, adding another layer of complexity and confusion when trying to perform this task.

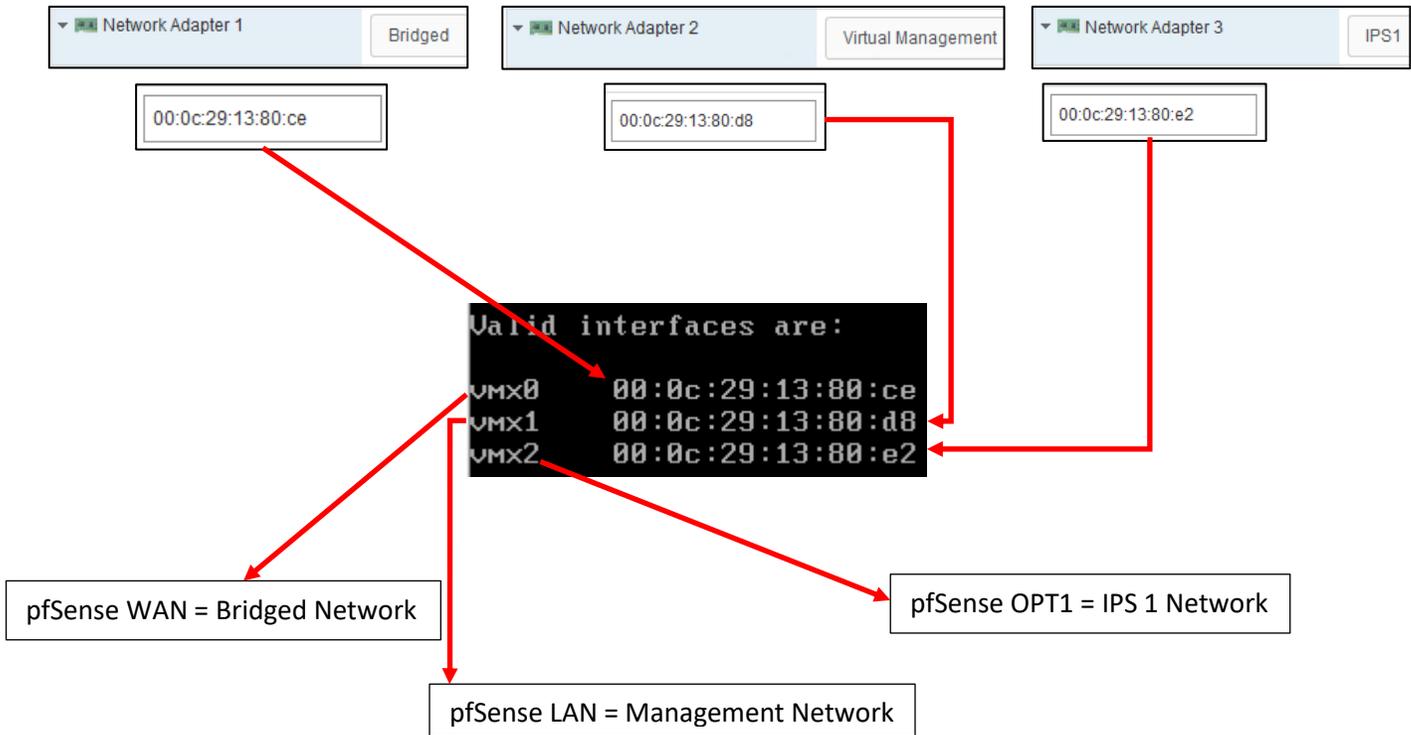
pfSense itself is based on the FreeBSD operating system, and BSD has its own methods for assigning physical (or virtual, in our case) network interfaces an interface name. For example, BSD assigned the network adapters of my virtual machine the interface names *vmx0*, *vmx1*, and *vmx2*. Every network adapter – integrated or not, virtual or physical, wired or wireless – all have a MAC address to uniquely identify them on a local network. We're going to take advantage of that to know for certain which of the three interfaces *vmx0* through *vmx2* map to *network adapters 1* through *3*, and how they should be assigned as the *WAN*, *LAN* and *OPT1* aliases. Students were highly advised to record the contents of the *Mac Address* input boxes of all three network adapters to assist in this task.

A quick way for readers to determine the interface names for their pfSense installation is through the wizard itself. A section of text labeled *Valid interfaces are* appears, followed by a series of lines. **Students should have 3 of these lines in total.** These lines provide the interface names, MAC addresses, current operational status, and type of hardware BSD identifies the network interface as (The drivers BSD loaded) for each network interface pfSense was able to detect. Here is an example:

```
Valid interfaces are:
 1          2
vmx0      00:0c:29:13:80:ce (down) VMware VMXNET3 Ethernet Adapter
vmx1      00:0c:29:13:80:d8 (down) VMware VMXNET3 Ethernet Adapter
vmx2      00:0c:29:13:80:e2 (down) VMware VMXNET3 Ethernet Adapter
```

13-50: A portion of the *Assign Interfaces* wizard. Pay attention to the interface names (1) and the MAC addresses for those interface names (2). This information is needed to determine which virtual network segment they are connected to. This in turn allows students to assign the *WAN*, *LAN* and *OPT1* interfaces correctly.

Compare the MAC addresses displayed, to the MAC addresses recorded earlier, and use that information to complete the rest of the Assign Interfaces wizard. A diagram (fig. 13-51) is provided below to help students understand how to correctly perform this mapping process.



13-51: Here we have the network configuration for my pfSense VM, and the output from the valid interfaces table from the *Assign Interfaces* wizard. Network adapter 1 (Bridged port group) has the MAC Address 00:0C:29:13:80:CE. Looking at the valid interfaces table, vmx0 has the same MAC address, that means vmx0 should be assigned as the WAN interface. The MAC address of the adapter attached to the Virtual Management port group (Network Adapter 2) matches the MAC address for vmx1. This means vmx1 should be assigned the LAN interface. Finally, the adapter connected to the IPS1 port group (Network Adapter 3) matches the MAC address for vmx2. This means that vmx2 should be assigned the OPT1 interface.

The remainder of this section will aim to guide students through the various questions the wizard will ask (in *italicized* font), and the answers I provided (in **bold** font) based on my lab network and adapter to MAC address mappings. **Students should be aware that this is by and far the most important configuration task for pfSense.** Making sure that the network adapters map to the correct pfSense aliases and port groups is absolutely vital to the lab environment working correctly.

Should VLANs be set up now [y|n]? n

*Enter the WAN interface name or 'a' for auto-detection
(vmx0 vmx1 vmx2 or a): **vmx0***

*Enter the LAN interface name or 'a' for autodetection
NOTE: this enables full Firewalling/NAT mode.
(vmx1 vmx2 a or nothing if finished): **vmx1***

*Enter the Optional 1 interface name or 'a' for auto-detection
(vmx2 a or nothing if finished): **vmx2***

The interfaces will be assigned as follows:

WAN -> vmx0

LAN -> vmx1

OPT1 -> vmx2

Do you want to proceed [y|n]? y

After answering these questions, pfSense will bring students to the command-line menu. This interface consists of a series of options, numbered one through sixteen that users can access by inputting the number of the option they desire.

```

Network interface mismatch -- Running interface assignment option.
vnx0: link state changed to UP
vnx1: link state changed to UP
vnx2: link state changed to UP

Valid interfaces are:

vnx0      00:0c:29:13:80:ce (down) VMware VMXNET3 Ethernet Adapter
vnx1      00:0c:29:13:80:d8 (down) VMware VMXNET3 Ethernet Adapter
vnx2      00:0c:29:13:80:e2 (down) VMware VMXNET3 Ethernet Adapter

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y:n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(vnx0 vnx1 vnx2 or a): vnx0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(vnx1 vnx2 a or nothing if finished): vnx1

Enter the Optional 1 interface name or 'a' for auto-detection
(vnx2 a or nothing if finished): vnx2

The interfaces will be assigned as follows:

WAN   -> vnx0
LAN   -> vnx1
OPT1  -> vnx2

Do you want to proceed [y:n]? y

Bootup complete

FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

VMware Virtual Machine - Netgate Device ID: b6cffe04a1f98f0c9e4

*** Welcome to pfSense 2.4.5-RELEASE-p1 (amd64) on pfSense ***

WAN (wan)      -> vnx0      -> v4/DHCP4: 10.0.0.26/24
                v6/DHCP6: 2601:408:502:c330:20c:29ff
/64
LAN (lan)      -> vnx1      -> v4: 192.168.1.1/24
OPT1 (opt1)    -> vnx2      ->

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults    13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option:

```

13-52: A selection of screen captures from the *Assign Interfaces* wizard, stitched together to show the questions the wizard asks, and the responses based on network adapter mappings in *fig. 13-51*. This is what students see upon first booting into pfSense. When finished, students are greeted with the pfSense command-line menu.

13.6.4.2 Setting IP Addresses for WAN, LAN, and OPT1

The next task to perform on the pfSense command-line is assigning IP addresses to the *WAN*, *LAN*, and *OPT1* interfaces using the *Set interface(s) IP address* wizard. Provided are connected to a network where DHCP is available, and they followed the guidance on configuring a static DHCP mapping for the pfSense virtual the *WAN* interface should already have an IP address, subnet mask, default gateway (and usually, DNS servers to forward DNS requests to) automatically provided (if this is not the case, see the sidebar discussion, *Help! The WAN Interface has no IP Address*, for some troubleshooting pointers). That means we should only have to run through the *Set interface(s) IP address* wizard twice – once for the *LAN* interface, and once for the *OPT1* interface. Select option 2 from the pfSense menu to get started.

Similar to the previous section (13.6.4.1), the remainder of this section is going to consist of the questions the *Set interface(s) IP address* wizard will ask students (*italicized*), and the correct answers for the *LAN* and *OPT1* interfaces (in **bold**), followed by an illustration depicting the same questions and answers.

LAN interface:

Available interfaces:

1 – WAN (*[interface name] – [dhcp/dhcp6/static address configuration]*)

2 – LAN (*[interface name] – static*)

3 – OPT1 (*[interface name]*)

Enter the number of the interface you wish to configure: **2**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.1.1**

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense

e.g. 255.255.255.0 = 24

255.255.0.0 = 16

255.0.0.0 = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**
Enter the start address of the IPv4 client address range: **172.16.1.10**
Enter the end address of the IPv4 client address range: **172.16.1.254**
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

Please wait while the changes are saved to LAN...

Reloading filter...

Reloading routing configuration...

DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24

You can now access the webConfigurator by opening the following URL in your web browser:

<https://172.16.1.1>

Press <ENTER> to continue. <ENTER>

```
Enter the number of the interface you wish to configure: 2
Enter the new LAN IPv4 address. Press <ENTER> for none:
> 172.16.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
    255.255.0.0   = 16
    255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.1.10
Enter the end address of the IPv4 client address range: 172.16.1.254
Disabling IPv6 DHCPD...
Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 172.16.1.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
    https://172.16.1.1/

Press <ENTER> to continue.
```

13-53: Screen captures from the *Set interface(s) IP address wizard*, stitched together to show the questions the wizard asks, and the responses for the LAN interface based on network adapter mappings in *fig. 13-51*.

Here is an abridged set of *questions* and **answers** for the *OPT1* interface:

OPT1 interface (abridged):

Available interfaces:

- 1 – WAN ([*interface name*] – [*dhcp/dhcp6/static address configuration*])
- 2 – LAN ([*interface name*] – *static*)
- 3 – OPT1 ([*interface name*])

Enter the number of the interface you wish to configure: **3**

Enter the new LAN IPv4 address: Press <ENTER> for none:

> **172.16.2.1**

Enter the new LAN IPv4 subnet bit count (1 to 31):

> **24**

For WAN, enter the new LAN IPv4 upstream gateway address.

For a LAN, press <ENTER> for none:

> **<ENTER>**

Enter the new LAN IPv6 address. Press <ENTER> for none:

> **<ENTER>**

Do you want to enable the DHCP server on LAN? (y/n) **y**

Enter the start address of the IPv4 client address range: **172.16.2.10**

Enter the end address of the IPv4 client address range: **172.16.2.254**

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) **n**

```

Enter the number of the interface you wish to configure: 3

Enter the new OPT1 IPv4 address. Press <ENTER> for none:
> 172.16.2.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0    = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 24

For a WAN, enter the new OPT1 IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Enter the new OPT1 IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on OPT1? (y/n) y
Enter the start address of the IPv4 client address range: 172.16.2.10
Enter the end address of the IPv4 client address range: 172.16.2.254

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to OPT1...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 OPT1 address has been set to 172.16.2.1/24

Press <ENTER> to continue.

```

13-54: Screen captures from the *Set interface(s) IP address* wizard, stitched together to show the questions the wizard asks, and the responses for the *OPT1* interface based on network adapter mappings in *fig. 13-51*.

After running the wizard again for the *OPT1* interface, students should have an IP address for the *WAN*, *LAN* and *OPT1* interfaces. Additionally, DHCP ranges should be assigned for the *LAN* and *OPT1* interfaces. We're just about ready to move to the webConfigurator, but before doing so, lets run some network connectivity tests first.

```

WAN (wan)      -> vmx0      -> v4/DHCP4: 10.0.0.26/24
                v6/DHCP6: 2601:408:502:c330:20c:29ff:fe13:80ce
/64
LAN (lan)     -> vmx1      -> v4: 172.16.1.1/24
OPT1 (opt1)   -> vmx2      -> v4: 172.16.2.1/24

```

13-55: The interface information portion of the pfSense command-line menu should look something like this. Looking good is one thing, now let's see if it actually works.

What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?

Unfortunately, I have no way of knowing what network ranges students use at home, so it's entirely possible your physical network may already be using one of the ranges I'm asking you to configure for your lab environment (e.g., 172.16.1.0/24, or 172.16.2.0/24). **To avoid network conflicts on your home network, maybe try these alternate configurations for the *Set interface(s) IP address wizard*:**

Alternate LAN configuration:

LAN interface IP address: 172.16.11.1
Subnet mask bit count: 24
DHCP start address: 172.16.11.10
DHCP end address: 172.16.11.254

Alternate OPT1 configuration:

OPT1 interface IP address: 172.16.12.1
Subnet mask bit count: 24
DHCP start address: 172.16.12.10
DHCP end address: 172.16.12.254

If your lab network is connected to a school or enterprise network using the entire 172.16.0.0/12 allocation, things may be a little more complicated. It may be best to use one of the other RFC1918 network allocations instead, such as 192.168.0.0/16, or 10.0.0.0/8. Why? Enterprise networking can become complicated, either due to growth over time, legacy configurations, or work-arounds to problems accrued over time. You don't want to troubleshoot network problems on your host system, nor do you want the IT ops team coming to your desk over a network outage that could've been avoided. **Here are some alternate configurations for the *Set interface(s) IP address wizard* if you need to avoid using 172.16.0.0/12 entirely:**

Alternate LAN configuration 1:

LAN interface IP address: 10.0.11.1
Subnet mask bit count: 24
DHCP start address: 10.0.11.10
DHCP end address: 10.0.11.254

Alternate OPT1 configuration 1:

LAN interface IP address: 10.0.12.1
Subnet mask bit count: 24
DHCP start address: 10.0.12.10
DHCP end address: 10.0.12.254

Alternate LAN configuration 2:

LAN interface IP address: 192.168.11.1
Subnet mask bit count: 24
DHCP start address: 192.168.11.10
DHCP end address: 192.168.11.254

Alternate OPT1 configuration 2:

LAN interface IP address: 192.168.12.1
Subnet mask bit count: 24
DHCP start address: 192.168.12.10
DHCP end address: 192.168.12.254

Substituting Instructions for Your Chosen Network Ranges

Keep in mind you don't have to use the alternate configurations recommended above. If students have some experience with networking and subnetting, they're welcome to use any network range that suits them. These are just some suggestions to help those who are not quite as experienced, and want to avoid network conflicts.

As a final reminder, **the remaining sections, chapters, and configuration steps will all assume that readers are using 172.16.1.0/24 for the LAN network and 172.16.2.0/24 for the OPT1 network.** This means you will have to mentally substitute steps and commands for the network range you are using instead.

For example, the lab network diagram in chapter 6 has the Kali VM on the IPS 1 (OPT1) network, with an IP address of 172.16.2.2. If you are using an alternate network configuration for the OPT1 network, say 192.168.12.0/24, then the Kali VM's IP address should be 192.168.12.2. If I say "*run the command ssh username@172.16.2.2 to connect to the kali VM*", you'll have to mentally substitute that with `ssh username@192.168.12.2` instead. As another example, firewall rules denying access to or from 172.16.2.3 (Metasploitable2) should be created for 192.168.12.3 instead. Keep this in mind as you continue to build your lab network!

Help! The WAN Interface has no IP Address

If the WAN interface of your pfSense VM has no IP address, consider some of the following to help with troubleshooting:

-No DHCP – It's pretty rare, but perhaps the WAN interface is bridged to a network without DHCP. This just means that you'll have to run the *Set interface(s) IP address* wizard to manually configure the WAN interface IP address, subnet mask, and default gateway. I've already listed the questions the wizard asks, and provided the answers for the LAN and OPT1 interfaces, but since I have absolutely no idea what IP address range and subnet mask are assigned to your local physical network, I cannot tell you what you need to enter for the wizard.

If you don't know either, ask a network administrator or whoever is responsible for your network to assist you. Note that if required to manually configure these settings here, practically all of the tasks that require DNS to be configured (e.g., network connectivity tests, and checking for updates on your VMs) will not work until DNS server addresses are configured. This can be done via the webConfigurator, and will be covered shortly.

-NAC Interference – If you're network security enthusiast at home or connected to an enterprise network, NAC (network access control) may be preventing the WAN interface from obtaining an IP address. Unfortunately, ESXi is NOT a hosted hypervisor, so the hack I've shown you for hosted hypervisors (e.g., using NAT and sharing the hypervisor host's IP address) isn't going to work here. The only way to deal with NAC interference on a bare-metal hypervisor is to talk to the IT or Security staff for your network, and see if you can get an exception for your ESXi server.

-Incorrect Virtual Switch/Port Group/NIC Configuration – Another possibility is perhaps the WAN network adapter is connected to the wrong port group and/or virtual switch. Refer to [section 13.5.2](#) (pp. 553-560). Double check that the virtual switches and port groups are configured correctly, and that the *Bridged* virtual switch has an uplink with an active connection. Make sure to double check that the correct interface on the pfSense VM was assigned as the WAN interface. Check out [section 13.6.3](#) (pp. 579-581) for a refresher on checking the MAC addresses of each network interface on the pfSense VM, and [section 13.6.4.1](#) (pp. 583-586) for the process on mapping the interfaces correctly on the pfSense VM.

13.6.5 Testing Internet Connectivity using Shell commands

Select option 8, labeled *Shell* in the pfSense menu. Doing so will open up a command-line (bash) shell. Run these 3 commands, and observe their output:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

Here is output from these 3 commands:

```
Enter an option: 8

[2.4.5-RELEASE][root@pfSense.localdomain]/root: ping -c 4 www.google.com
PING www.google.com (172.217.6.100): 56 data bytes
64 bytes from 172.217.6.100: icmp_seq=0 ttl=54 time=24.496 ms
64 bytes from 172.217.6.100: icmp_seq=1 ttl=54 time=22.714 ms
64 bytes from 172.217.6.100: icmp_seq=2 ttl=54 time=21.638 ms
64 bytes from 172.217.6.100: icmp_seq=3 ttl=54 time=19.490 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 19.490/22.084/24.496/1.813 ms
[2.4.5-RELEASE][root@pfSense.localdomain]/root: nslookup www.google.com
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.6.100
Name:   www.google.com
Address: 2607:f8b0:4009:812::2004

[2.4.5-RELEASE][root@pfSense.localdomain]/root: curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Mon, 01 Jun 2020 02:53:41 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Mon, 01 Jun 2020 02:53:41 GMT
cache-control: private
set-cookie: 1P_JAR=2020-06-01-02; expires=Wed, 01-Jul-2020 02:53:41 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=DdNU16afHrYu25Utm83temwvvrSe6a4UyA3YHz_JKLFzBAv7xrWi8HjSn2-x1PNmxh3EutjAoFBh15hNpxrU72.jpzLLQU0JHJxaOMh5mFyntk5Gae7KUMe2-d1g8I1KloIb7HzOBP_BB4b0sb4lt0Tv1zwOdriUE8ndqfygcrN04; expires=Tue, 01-Dec-2020 02:53:41 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
```

13-56: The output from the `ping -c 4`, `nslookup`, and `curl -I` commands. All three of these commands completed successfully. Pay close attention to the marked sections above. Note that the IP addresses returned for `nslookup` (In the fields labeled *Address*) may vary based on region.

In a nutshell, these three commands are being used to test various forms of internet connectivity for our VM. `ping -c 4 www.google.com` tells pfSense to send 4 (and only 4) ICMP packets to a specific destination, requesting that the destination respond with its own ICMP packets if it has been reached. `nslookup www.google.com` asks our pfSense virtual machine's configured DNS servers to translate a domain name to an IP address for us. Finally, `curl -I https://www.google.com` is being used to test HTTPS connectivity to the internet. The `-I` option tells the command to only return the HTTP Server headers from our request. All we're really interested in is the line of text: *HTTP/2 200*. This is a thumbs up from Google's webserver confirming that they got our HTTP request with no problems.

Students already familiar with DNS basics may have noticed that we are already trying to ping a domain name (`www.google.com`) with our `ping` command. This means, that in order to actually ping the correct destination, our virtual machine will need to make a DNS request to find the IP address of `www.google.com`. That makes the `nslookup` test redundant, right? Well, yes and no. Later in this chapter, as new virtual machines get created, readers will be advised to perform connectivity tests on those VMs as well. However, the pfSense firewall policy is going to be very strict, so ICMP packets outbound from our lab network will be blocked. Due to how DNS works, the `nslookup` check can still be used to make sure VMs can resolve domain names, and the `curl` connectivity test will be more than sufficient to confirm whether or not lab virtual machines have the internet access they require.

My connectivity commands failed! Now what?

If students got anything other than output similar to *fig. 13-56* (e.g., request timeouts and/or packet loss for `ping`, timeouts for `nslookup`, no response for `curl -I`), then there are connectivity issues to be sure. Troubleshooting network connectivity is an extremely complex topic. I can't give you a definitive guide for finding the root of your problem, but I can tell you to start with the basics and work your way up – sometimes the cause of your network problems are settings or hardware that was taken for granted.

Checking physical cabling, link lights and physical connectivity to network devices always comes first. As an extension to that, check out the sidebar in section 13.6.4.2 (*Help! The WAN interface has no IP address*) for some additional clues. The VM may be bridged to the wrong port group, etc. Some form of network security (e.g., a network firewall) may be preventing your VM from connecting to the internet.

If students were required to run the *Set interface(s) IP address* wizard for the *WAN* interface (No DHCP), or your local network's DHCP server doesn't assign DNS servers automatically, your troubleshooting commands will fail because pfSense has no way of resolving domain names. We will be covering how to manually configure a primary and/or secondary DNS server for pfSense via the webConfigurator shortly.

If your host system is connected to a physical network already using 172.16.1.0/24 or 172.16.2.0/24, you may be experiencing network conflicts, routing loops, or other weird behavior. Assign different IP addresses and ranges to the *LAN* and *OPT1* networks to avoid network conflicts. See the sidebar discussion in 13.6.4.2 labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?*

Last but not least, check and double check that you entered the commands correctly. Typos matter on the command-line, and BSD will not hold your hand if the command is entered incorrectly. If all else fails, don't be afraid to ask others for guidance.

13.6.5.1 One Last Detail (enableallowallWAN)

Before exiting the command line interface (or, if students already exited the shell, select option 8 from the menu again to open new session) **there is a command that students must run in order to gain initial access to the pfSense WebConfigurator** (aka the web interface for pfSense):

```
pfSsh.php playback enableallowallwan
```

This command runs a script that disables some of the default firewall rules that ships with pfSense. One of these rules blocks all traffic from all RFC1918 IP addresses (e.g., 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8) attempting to access the WAN interface. Under normal circumstances, this is a nice, solid, secure default setting. pfSense operates on the assumption that users will attempt to access its web interface (and administrative tools) from the LAN network, and LAN interface only. However, students running pfSense on ESXi will be required to access the pfSense web interface through its WAN interface, instead. So, the default firewall rule blocking that access must be disabled. That's where this command comes into play. After running it, students may type exit to leave the shell.

```
[2.4.5-RELEASE][root@pfSense.localdomain]/root: pfSsh.php playback enableallowallwan
Adding allow all rule...
Turning off block private networks (if on)...
Turning off block bogon networks (if on)...
Reloading the filter configuration...
[2.4.5-RELEASE][root@pfSense.localdomain]/root: exit
```

13-57: Select option 8, run `pfSsh.php playback enableallowallwan`, then exit the shell. **This task must be performed before continuing to chapter 14.**

Run and Gun

Be aware that after running this command, anyone will be able to attempt to log in to the webConfigurator of your pfSense virtual machine, provided they know its IP address. Therefore, **I recommend connecting to the pfSense webConfigurator immediately after running this command and going through the initial setup wizard in order to change the default password that pfSense ships with.** From there, complete the rest of chapter 14 in order to restrict access to the webconfigurator to just your management workstation (or jump box).

13.6.6 Finish setting up pfSense

Navigate to [chapter 14, pfSense Firewall Policy and Network Services](#), starting on *p.* 664 and follow the chapter guidance. Once completed, readers will be directed back here to complete their lab environment.

13.7 Create the Remaining Virtual Machines

Welcome back! Now that the pfSense VM is fully functional, it's time to start working on the remaining lab VMs. In this section, users will create three of the four remaining virtual machines via the *New Virtual Machine* wizard, then adjust the *Settings* of each virtual machine. After the SIEM, IPS and Kali VMs are created and configured, readers will be guided through the operating system installation, and initial setup process for all three VMs. The Metasploitable 2 VM is a unique case, and will be covered separately.

13.7.1 Virtual Machine Creation and Tuning – SIEM, IPS and Kali

Run the *New Virtual Machine* wizard three times, with the settings listed below. Assume the default for any settings not mentioned in the table below. Refer back to [section 13.6.1](#) (pp. 569-574) for guidance on how to access and progress through the wizard as needed.

Name:	SIEM	IPS	Kali
Guest OS Family:	Linux	Linux	Linux
Guest OS Version:	Ubuntu Linux (64-bit)	Ubuntu Linux (64-bit)	Debian GNU/Linux 11 (64-bit)
Storage:	Use any available datastore	Use any available datastore	Use any available datastore
Memory:	4GB (4096MB)	4GB (4096MB)	4GB (4096MB)
Hard Disk 1:	80GB Thick Provisioned, eagerly zeroed	80GB Thick Provisioned, eagerly zeroed	80GB Thick Provisioned, eagerly zeroed
Number of Network Adapters:	1	3 (Click <i>Add network adapter</i> twice)	1
Port Groups:	Network Adapter 1: Virtual Management	Network Adapter 1: Virtual Management Network Adapter 2: IPS1 Network Adapter 3: IPS2	Network Adapter 1: IPS1
CD/DVD Drive 1:	Select <i>datastore ISO file</i> . Locate the Ubuntu Server ISO, and select it.	Select <i>datastore ISO file</i> . Locate the Ubuntu Server ISO, and select it.	Select <i>datastore ISO file</i> . Locate the Kali Linux ISO, and select it.
Other:	Remove <i>USB controller 1</i>	Remove <i>USB controller 1</i>	Remove <i>USB controller 1</i>

Name	SIEM
Datastore	Mune
Guest OS name	Ubuntu Linux (64-bit)
Compatibility	ESXi 7.0 virtual machine
vCPUs	1
Memory	4096 MB
Network adapters	1
Network adapter 1 network	Virtual Management
Network adapter 1 type	VMXNET 3
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SCSI controller 0	LSI Logic Parallel
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	80GB
Datastore	[Mune] SIEM/
Mode	Dependent
Provisioning	Thick provisioned, eagerly zeroed
Controller	SCSI controller 0 : 0
CD/DVD drive 1	
Backing	[Flash_Step] ISOs/ubuntu-20.04.1-live-server-amd64.iso
Connected	Yes

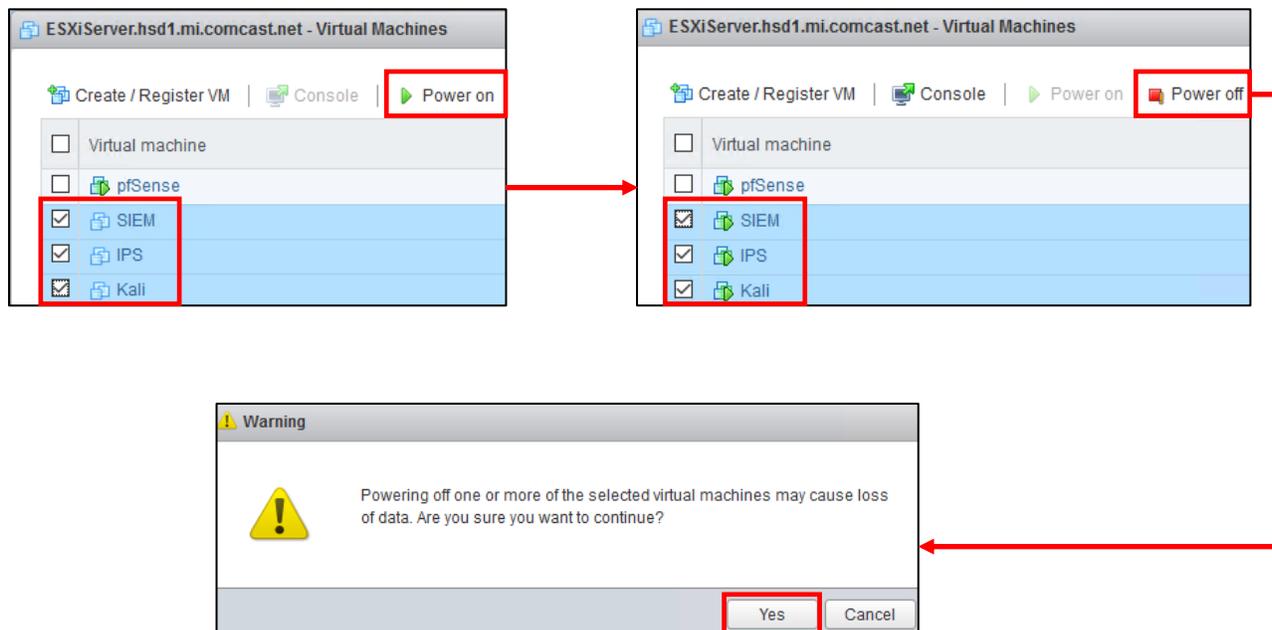
Name	Kali
Datastore	Masa
Guest OS name	Debian GNU/Linux 11 (64-bit)
Compatibility	ESXi 7.0 virtual machine
vCPUs	1
Memory	4096 MB
Network adapters	1
Network adapter 1 network	IPS1
Network adapter 1 type	VMXNET 3
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SCSI controller 0	VMware Paravirtual
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	80GB
Datastore	[Masa] Kali/
Mode	Dependent
Provisioning	Thick provisioned, eagerly zeroed
Controller	SCSI controller 0 : 0
CD/DVD drive 1	
Backing	[Flash_Step] ISOs/kali-linux-2020.3-installer-amd64.iso
Connected	Yes

Name	IPS
Datastore	Masa
Guest OS name	Ubuntu Linux (64-bit)
Compatibility	ESXi 7.0 virtual machine
vCPUs	1
Memory	4096 MB
Network adapters	3
Network adapter 1 network	Virtual Management
Network adapter 1 type	VMXNET 3
Network adapter 2 network	IPS1
Network adapter 2 type	VMXNET 3
Network adapter 3 network	IPS2
Network adapter 3 type	VMXNET 3
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SCSI controller 0	LSI Logic Parallel
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	80GB
Datastore	[Masa] IPS/
Mode	Dependent
Provisioning	Thick provisioned, eagerly zeroed
Controller	SCSI controller 0 : 0
CD/DVD drive 1	
Backing	[Flash_Step] ISOs/ubuntu-20.04.1-live-server-amd64.iso
Connected	Yes

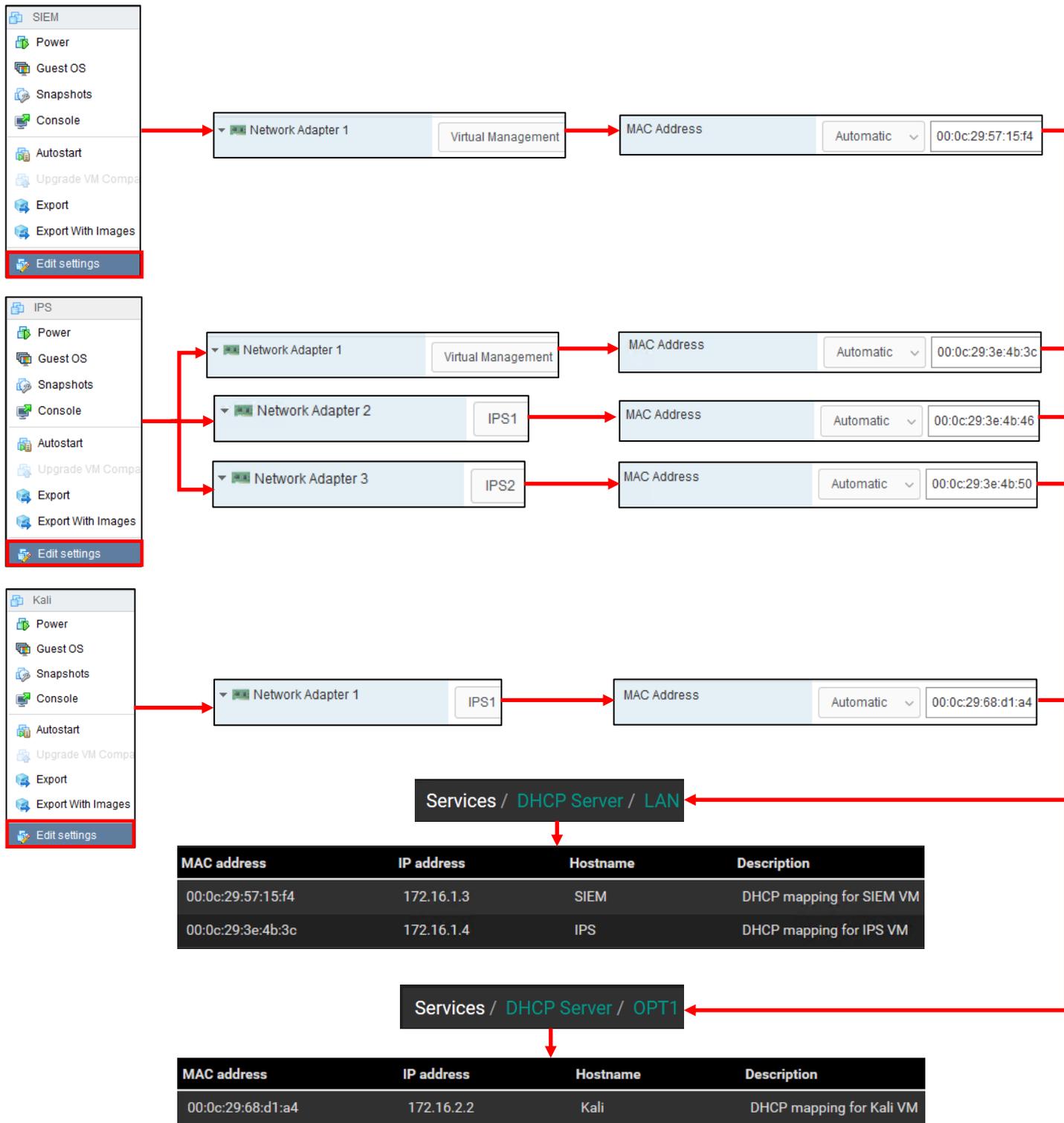
13-58: The summary page for the SIEM, IPS, and Kali virtual machines. When students are finished, the summary screens for all three virtual machines should look similar to the illustration above.

Before we proceed, there's one more adjustment students need to make to their virtual machines: the SIEM, IPS, and Kali VMs all need to be powered on and powered off again. Recall in section 13.6.3 (specifically, the sidebar discussion, *MAC and Cheese* on p. 579) that ESXi doesn't assign MAC addresses to network adapters until after the virtual machine is first booted. The easiest way for Students to proceed is to power on, then immediately turn off all three virtual machines, then record the MAC addresses of each virtual adapter, the virtual machine they belong to, and the port group they are attached to.

From there, students will log into the pfSense VM and configure static DHCP allocations for the SIEM VM, Kali VM, and the network adapter attached to the *Virtual Management* port group of the IPS VM. The SIEM VM should be statically assigned the IP address 172.16.1.3, the Management Virtual Switch of the IPS VM should be assigned 172.16.1.4. Both of these allocations should be configured on the *LAN* interface of the IPS VM. Meanwhile, the Kali VM should be assigned the IP address 172.16.2.2 on the *OPT1* interface. Students may refer to Chapter 14, section 14.3.4.1 (pp. 690-692) for a refresher on creating static DHCP mappings on pfSense.



13-59: Turn the SIEM, IPS, and Kali VMs on and off again to force ESXi to assign MAC addresses to the network adapters. The fastest way to accomplish this task is to navigate to the *Virtual Machines* listing (*Navigation > Virtual Machines*), click the checkbox next to the SIEM, IPS, and Kali virtual machines, then click the *Power on* button. Then, after a moment or two, with the same virtual machines selected, click the *Power off* button.

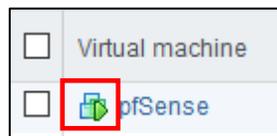


13-60: With the SIEM, IPS and Kali VMs powered off, open up their individual settings menus and record the MAC addresses for each interface, noting which virtual switch its attached to (check out the *Noting the Notable* sidebar conversation on p. 581 for a nice template to use for documenting the lab VMs). Use the MAC addresses and create three static DHCP mappings on the pfSense webConfigurator. Create two mappings on the *LAN* interface for the SIEM VM, and the *Virtual Management* port group interface of the IPS VM, and a third mapping on the *OPT1* interface for the Kali VM. Be sure to *Apply Changes* on the pfSense webConfigurator to save these new DHCP mappings.

13.7.2 Operating System Installation

In this section, students will learn how to install the operating system for the SIEM, IPS, and Kali virtual machines. Both the SIEM and IPS VMs will have Ubuntu Server installed as their operating system, while the Kali VM will have the latest version of the Kali Linux distribution installed. The installation instructions will differ for each virtual machine, so please pay attention.

As a general reminder, please make sure that the pfSense VM is running, and that you have completed chapter 14 to ensure pfSense is ready to support the rest of the lab environment. Without the pfSense VM, none of the virtual machines will have internet access. That may result in the operating system installers failing in different ways. To confirm the pfSense is running, students can check tiny icon next to the name of the VM on the Virtual Machines pane. If the VM is running, A small green arrow will appear over a set of three overlapping blue squares next to the VM.



13-61: This little icon is a quick indicator that the VM is running. Running VMs will also have a value greater than zero for the *Host CPU* and *Host memory* columns on the *Virtual Machines* pane.

13.7.2.1 Installing Ubuntu on the SIEM VM

To get started, on the *Virtual Machines* pane, click the checkbox next to the SIEM entry in the virtual machine list to highlight it, then click the icon that looks like a play button underneath the list of virtual machines (just like with the pfSense VM). The virtual machine will begin booting off the Ubuntu Server ISO. The first screen students see will ask to confirm the language they wish to use. The default language should be *English*, so hit the enter key on your keyboard to continue.

Depending on when students downloaded their copy of the Ubuntu Server ISO, and how frequently the ISO is updated, a screen may appear titled *Installer update available*. This screen provides users with the option to download the latest version of the Ubuntu installation wizard, called Subiquity. Use the arrow keys on your keyboard to highlight *Update to the new installer*, then hit enter.

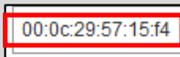
Note: If for some reason downloading the latest installer fails, there's a good chance that there are network problems with the lab environment elsewhere, and that there is troubleshooting to do. Students are welcome to select the *Continue without updating* option, but keep this in mind if the installer misbehaves or fails later. Check to see if the hypervisor host has internet connectivity, double check the firewall rules on the pfSense virtual machine, network settings, physical cabling, etc.

The next screen asks users to confirm their keyboard configuration. The default settings for both the *Layout* and *Variant* settings are *English (US)*. If you are not using a standard US-English keyboard, you may wish to use the arrow keys to highlight the *Identify keyboard* option, then hit enter. Otherwise, highlight *Done* on the bottom of the screen, and hit enter.

Next up, is the *Network connections* screen. a single network adapter should populate this page. The network adapter (named *ens160* in my case) should automatically be assigned the IP address 172.16.1.3. Below the IP address in light grey text is the MAC address of the network adapter that the Ubuntu installer detected. This should be the same MAC address of the network adapter of the SIEM VM. If the correct IP address was assigned, students can hit the enter key to continue (The *Done* option should be highlighted by default). Otherwise, see the section *What Reservation?* for some troubleshooting tips.

What Reservation?

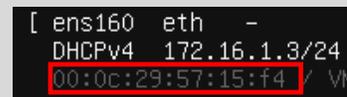
If for some reason the network adapter was assigned any other IP address other than 172.16.1.3, Refer back to section 13.7.1. Check the *MAC Address* field under *Network adapter 1* for the SIEM VM, in its *Edit settings* menu. Compare that MAC address to the MAC address used to create a static DHCP mapping on the *LAN* interface of the pfSense VM. Compare that to the MAC address displayed on the *Network connections* screen of the Ubuntu installer. **They should all be identical.** If there are any errors, correct them and reset the SIEM VM, to restart the Ubuntu installer until the pfSense DHCP assigns the network adapter the correct IP address.



00:0c:29:57:15:f4



MAC address
00:0c:29:57:15:f4



```
[ ens160 eth -  
DHCPv4 172.16.1.3/24  
00:0c:29:57:15:f4 / VM
```

13-62: If the SIEM VM failed to get the correct IP address, check the *Edit settings* menu of the virtual machine – specifically the *MAC Address* field under *Network Adapter 1*. Compare that to the MAC address used to create a static DHCP mapping on the *LAN* interface on the pfSense WebConfigurator. Correct the static DHCP entry as necessary then restart the SIEM VM to restart the ubuntu installer. Confirm that the network adapter was correctly assigned the 172.16.1.3 IP address.

The *Configure proxy* screen appears. Use the up arrow key to highlight the text box labeled *Proxy address* and enter `http://172.16.1.1:3128`. If you recall from Chapter 14, this is the IP address and port for the Squid proxy on the *LAN* interface of the pfSense VM. Use the arrow keys to highlight *Done*, and hit enter to continue.

The next screen, labeled *Configure Ubuntu archive mirror* will appear. This is another one of those situations where students will know whether or not they need to change this setting. Unless the lab environment is in an enterprise network and the network team happens to be operating their own software archive mirror, accept the default setting (in my case, the default mirror address was `http://us.archive.ubuntu.com/ubuntu`). With *Done* highlighted, hit enter to continue.

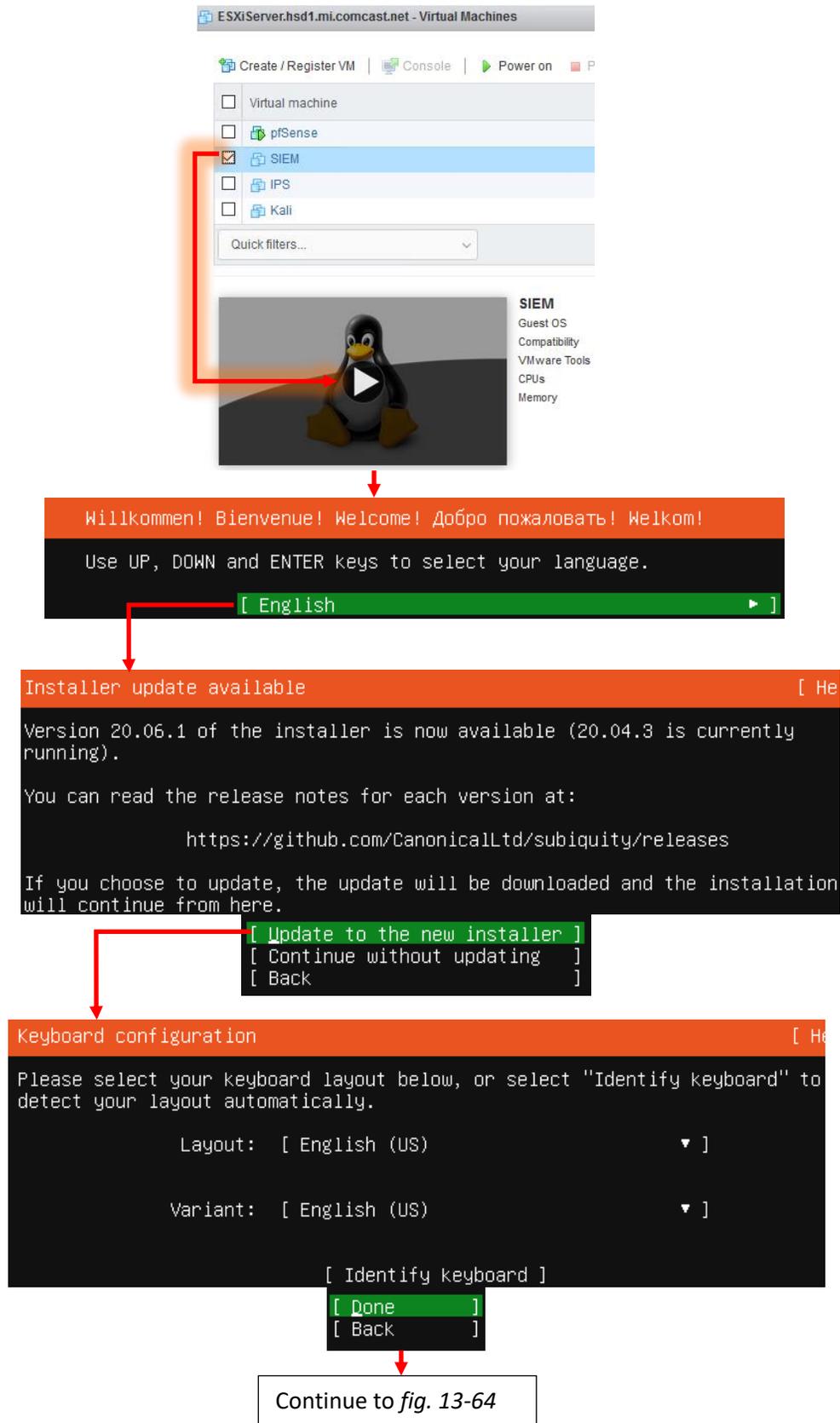
The *Guided storage configuration* screen appears next. Accept the default settings and let the Ubuntu installer format the entire disk. Use the arrow keys to highlight *Done*, and hit enter to continue. The next screen, titled *Storage configuration*, shows you how the installer is going to format the hard drive, and what partitions are going to be where in a large list labeled *FILE SYSTEM SUMMARY*. By default, *Done* should already be highlighted on this screen. If not, use the arrow keys to highlight it and hit enter to continue. A pop-up labeled *Confirm destructive action* appears. This screen informs the user that any data on the disk will be lost as a result of formatting and partitioning the disk. Since there is no data on the virtual hard disk yet, with *Continue* highlighted, hit the enter key to proceed.

Next is the *Profile setup* screen. There are five input boxes on this screen. Ubuntu asks the user for their name, the server's name, a username (that will be used to log in to the server later), the password for that username, followed by an input box asking the user to repeat the password. Students may enter any name, username, or password they would like, but it is recommended to both set the server name to *siem*, as well as to save the username and password combination to a password manager. Once finished, use the arrow keys to highlight *Done*, then hit enter to continue.

The *SSH Setup* screen appears and asks users if they would like to install the OpenSSH server package. By default, the prompt should be between two brackets next to the text *Install OpenSSH server*. Hit the spacebar to leave an 'X' between the brackets. Afterwards, use the arrow keys to highlight *Done* and hit enter to advance.

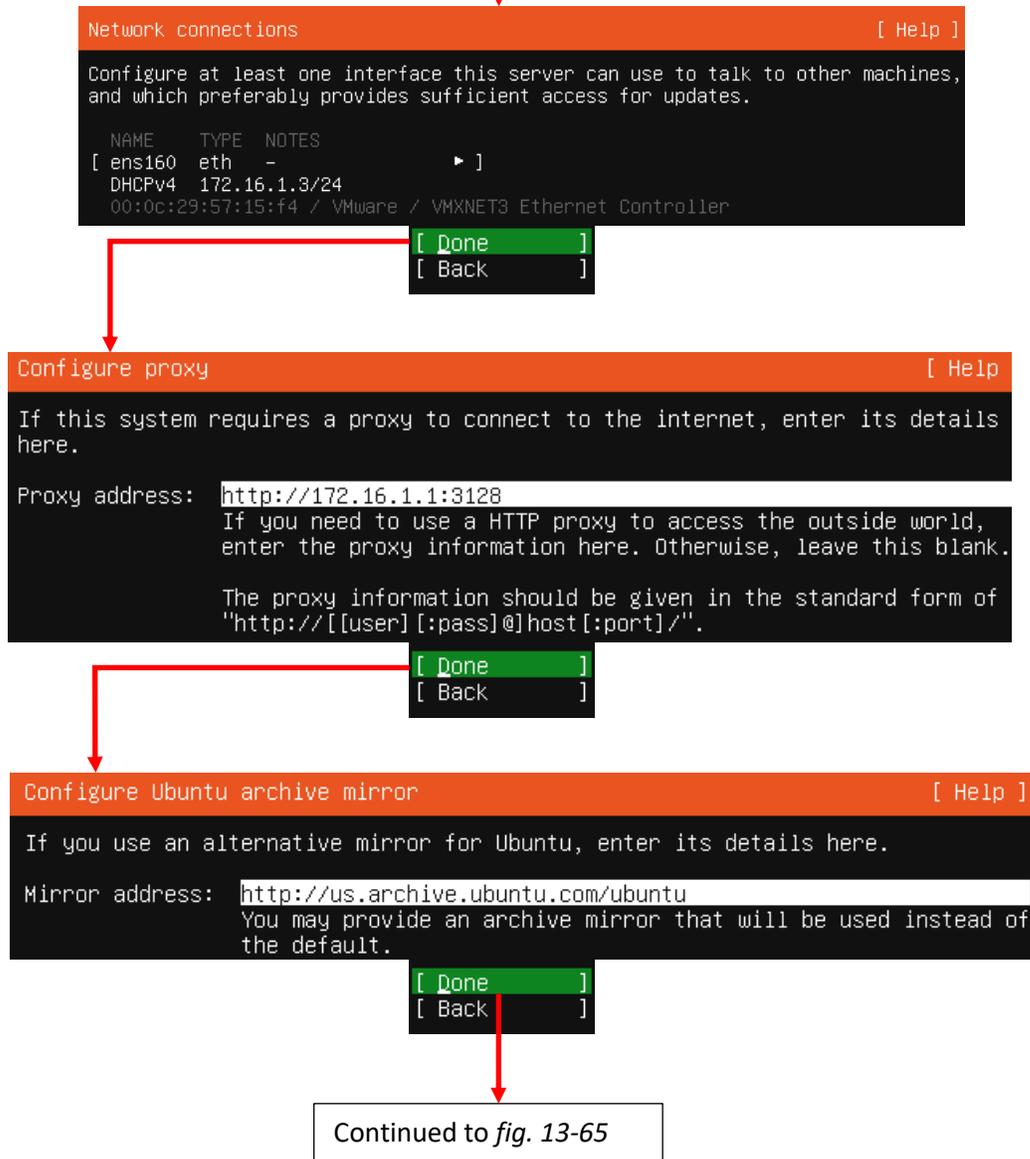
The next screen is labeled *Featured Server Snaps*. The latest versions of Ubuntu use an additional software manager called 'snap' to deliver software packages. Use either the arrow keys or the tab key to highlight *Done* and hit enter – Do not install any snaps, continue the installer.

We reach a new screen labeled *Install complete!* At this point, students have made all of the necessary decisions for the ubuntu installer to proceed, and handle all of the installation tasks at once. Once completed, the installer will grant students the option to reboot the system. However, instead of using the installer's reboot function, click the *Actions* option from the virtual console window, followed by *Power*, then *Power off* to shut down the virtual machine. Afterwards, close the virtual console.



13-63: On the Virtual Machines page, click the checkbox next to the SIEM VM, then click the play button under the virtual machines list. This powers on the VM and opens its virtual console. The user then selects the language of the installer, the installer checks for updates for itself, then asks the user to set the keyboard layout and variant language.

Continued from *fig. 13-63*



13-64: The next stages of the Ubuntu Server 20.04 installer. In these screens, students can confirm whether or not the static DHCP mapping for the SIEM VM is working correctly, configure the system to use the Squid proxy service configured on the pfSense VM, and confirm software archive mirror they would like to use.

Continued from *fig. 13-64*

```
Guided storage configuration
Configure a guided storage layout, or create a custom one:
(X) Use an entire disk
    [ /dev/sda local disk 80.000G ▼ ]
[X] Set up this disk as an LVM group
    [ Done ]
    [ Back ]
```

```
Storage configuration [ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT  SIZE  TYPE  DEVICE TYPE
[ /           39.498G new ext4 new LVM logical volume ▶ ]
[ /boot      1.000G new ext4 new partition of local disk ▶ ]

AVAILABLE DEVICES
DEVICE              TYPE              SIZE
[ ubuntu-vg (new)   LVM volume group  78.996G ▶ ]
free space
[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES
DEVICE              TYPE              SIZE
[ ubuntu-vg (new)   LVM volume group  78.996G ▶ ]
ubuntu-lv          new, to be formatted as ext4, mounted at / 39.498G ▶ ]
[ /dev/sda         local disk        80.000G ▶ ]
partition 1       new, bios_grub   1.000M ▶ ]
partition 2       new, to be formatted as ext4, mounted at /boot 1.000G ▶ ]
partition 3       new, PV of LVM volume group ubuntu-vg 78.997G ▶ ]

[ Done ]
[ Reset ]
[ Back ]
```

Continued to *fig. 13-66*

13-65: These screens are used to configure the storage settings for the operating system. Students will be using the default storage settings for the SIEM VM.

Continued from *fig. 13-65*

```
Confirm destructive action

Selecting Continue below will begin the installation process and
result in the loss of data on the disks selected to be formatted.

You will not be able to return to this or a previous screen once the
installation has started.

Are you sure you want to continue?

[ No ]
[ Continue ]
```

```
Profile setup [ Help ]

Enter the username and password you will use to log in to the system. You can
configure SSH access on the next screen but a password is still needed for
sudo.

Your name: ayy
Your server's name: siem
The name it uses when it talks to other computers.
Pick a username: ayy
Choose a password: *****
Confirm your password: *****

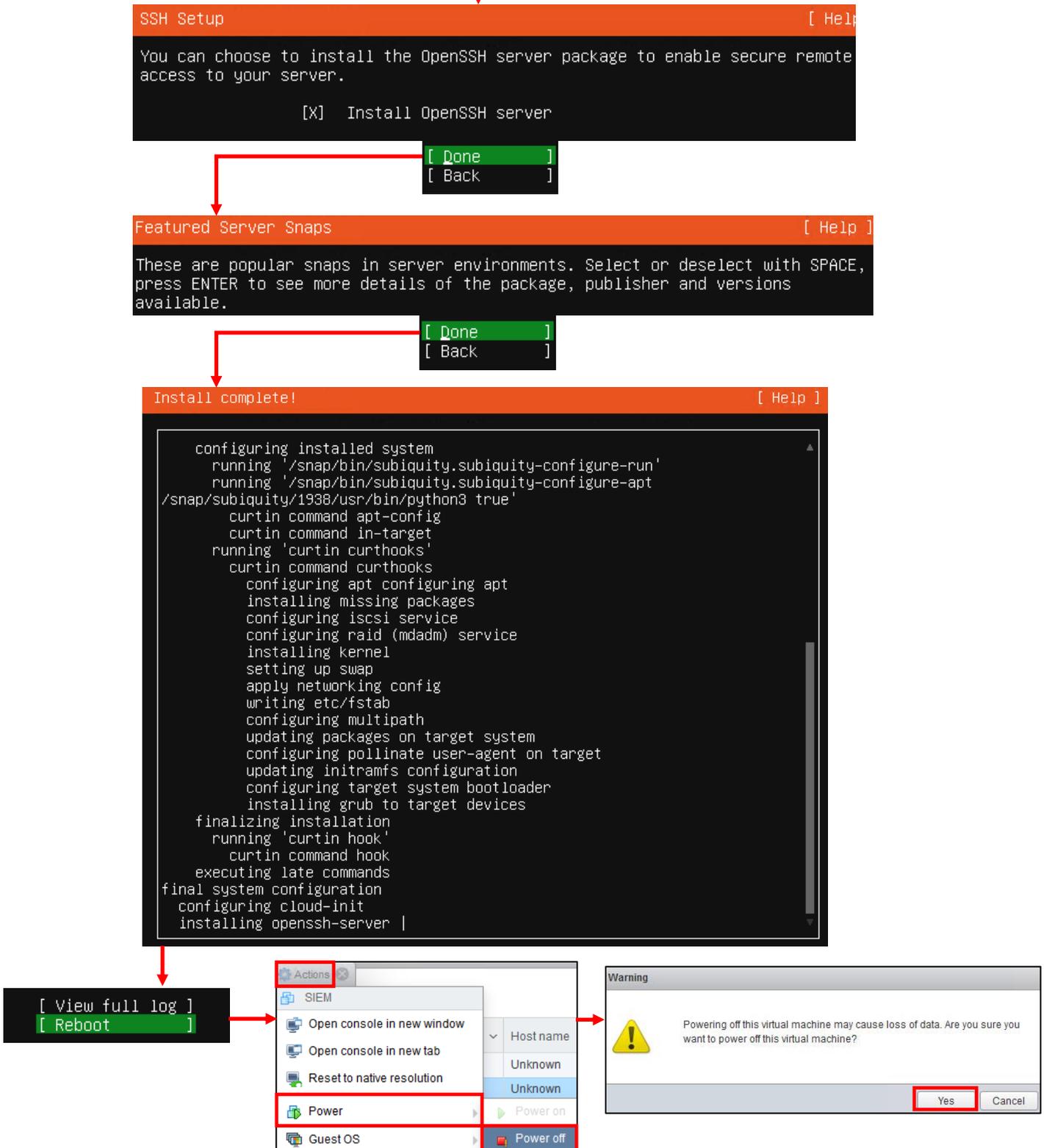
[ Done ]
```

Title:	SIEM VM
Username:	ayy
Password:	*****
URL:	172.16.1.3
<input type="checkbox"/> Expires:	7/21/2020 12:11 PM
<input checked="" type="checkbox"/> Notes:	User account credentials for the SIEM VM.

Continued to *fig. 13-67*

13-66: After confirming the storage configuration settings, users are prompted to name their server, and create a user account. It's recommended to store the username and password for the SIEM VM in a password manager.

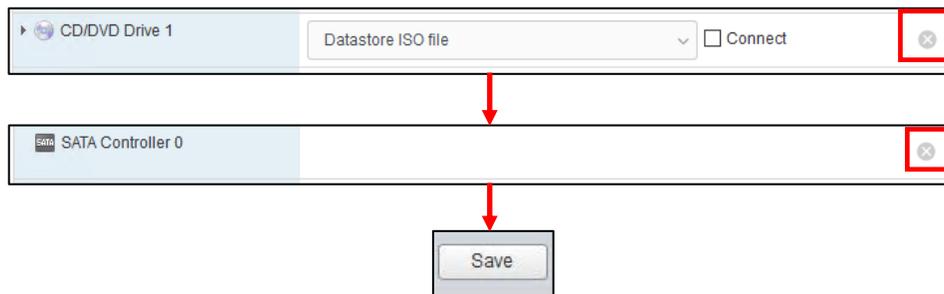
Continued from *fig. 13-66*



13-67: The final stages of the Ubuntu Server 20.04 installer. Select the option to install OpenSSH server, then decline to install any server snaps. Finally, once the installer grants you the option to reboot, ***Use the Actions menu on the virtual console to power off the virtual machine.***

13.7.2.2 Additional Virtual Machine Settings – SIEM VM

Now that the operating system installation is complete, there is one last configuration setting to adjust on the SIEM VM before we can boot into Ubuntu Linux and perform some diagnostic tasks. Back in [section 13.6.3](#) (pp. 579-581), students learned how to remove the *CD/DVD Drive 1* and *SATA Controller 0* virtual hardware from the pfSense VM. Students need to perform that task on the SIEM VM. Open the SIEM VM's *Edit settings* menu, locate the *DVD Drive*. Afterwards click the *Apply*, then *OK* buttons in the lower right corner to confirm this change, and close the SIEM VM's settings menu.



13-68: Access the SIEM VM's *Edit settings* menu and remove both the *CD/DVD Drive 1*, and *SATA Controller 0* virtual hardware. Recall that students may need to click *Save* after removing *CD/DVD Drive 1*, then re-open the menu to then remove *SATA Controller 0*. See the sidebar conversation, [What CD/DVD Drive](#) (p. 579).

13.7.2.3 Booting the SIEM VM for the first time

After changing the SIEM VM's settings, start the VM back up and bring up its virtual console. After a moment or two, you will be greeted with login prompt labeled *SIEM login*. Enter the username you configured during the installer, followed by the password to log in.

Some students may not be familiar with command-line applications, and that's okay. This is only a quick login to make sure network connectivity is working. Please type in the following commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The purpose of the `ip` command above is to display all of the network interfaces on the system. We pass this command the `-br` option for brief output, followed by the letter 'a' to indicate we're interested in seeing the IP addresses on our system. Users could replace 'a' with 'address' or 'addr' and the `ip` command would interpret it the same. We're using this command to serve as a secondary confirmation that the SIEM VM was successfully assigned the IP address 172.16.1.3, as displayed in [fig.13-69](#) below. Students may notice a second interface on the system designated `lo`. This is a "loopback" network interface and can safely be ignored.

The `nslookup` command is to confirm that the SIEM VM is able to resolve hostnames using DNS. The output from the command should be similar to what is presented in *fig. 13-69*. Finally, that brings us to the `curl` command. This command is to confirm connectivity to the internet over port 443, HTTPS. The `-I` option in the command tells `curl` to only return the headers from the web server being contacted. Once again, the output of this command should be fairly similar to what is presented in *fig. 13-69*.

```
Ubuntu 20.04 LTS siem tty1
siem login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@siem:~$ nslookup www.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:806::2004
ayy@siem:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Tue, 21 Jul 2020 20:10:38 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Tue, 21 Jul 2020 20:10:38 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-21-20; expires=Thu, 20-Aug-2020 20:10:38 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=1AAB5nk21PEgo8rGiFr-9PxEuTIY0Nx2tMMi-EmACtdRnP1PkB0xoosGu9FjWzbLyW0TG0HKUj6kLonn4Rr-yu-MIc8itYAI507X2Vvk2HzhKb1Wk0UK30rSk6Fd0ce6_Battd9PQ4YI7-CoRGf381n74m078YmTAAtz1XJf8-WM; expires=Wed, 20-Jan-2021 20:10:38 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
ayy@siem:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
ens160            UP                172.16.1.3/24 fe80::20c:29ff:fe57:15f4/64
```

13-69: After logging in to the SIEM virtual machine, students will need to run a series of network troubleshooting commands. These commands are to confirm that the SIEM VM has the correct IP address configured (`ip -br a`), can resolve hostnames via DNS (`nslookup www.google.com`), and has connectivity over HTTPS (`curl -I https://www.google.com`).

Before logging out of the SIEM virtual machine, there are three more commands to run, but before we can run them, we will need to become the root user. Enter the following command:

```
sudo su -
```

When prompted, enter the password for the user you created. If successful, you will be logged in as the root user on the SIEM virtual machine. The root user, sometimes referred to as the super user, is a special account that has complete authority over the system. Additionally, root has

access to special administrative commands that normal users are not allowed to use. As the root user, let's **run those last three commands in this exact order:**

```
apt-get update
apt-get -y dist-upgrade
init 6
```

Ubuntu is based off of the Debian Linux distribution. Because of this, it uses a package manager called apt (in addition to the snap package manager mentioned earlier). The two apt-get commands, apt-get update then apt-get -y dist-upgrade tell Ubuntu to reach out to the software archive mirror and get an updated list of software packages, then if any packages installed on the system need to be updated, updated them immediately. This set of commands also confirms that the Squid proxy server on the pfSense VM is working properly, and proxying all of the HTTP requests from the SIEM VM. The final command, init 6, tells the system to reboot immediately. As an alternative, users can also run the command reboot instead.

```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Fetched 317 kB in 1s (524 kB/s)
Reading package lists... Done
root@siem:~# apt-get -y dist-upgrade
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@siem:~# init 6_
```

13-70: the command `sudo su -` allows students to become the root user on the SIEM VM. We then use root's permissions to ensure all of the software packages on the system are up to date (`apt-get update`, followed by `apt-get -y dist-upgrade`), then immediately reboot the system (`init 6`, or optionally `reboot`). Be aware that the apt-get commands may take a little bit of time to finish, based on the number of updates available and speed of your internet connection.

Help! My apt-get commands are failing!

If you're experiencing problems with the apt-get commands failing to complete, there's a very good chance that the apt package manager is not properly configured to use the SQUID HTTP proxy we installed on the pfSense VM, or that the SQUID proxy service on the pfSense VM may be misconfigured. If students entered the wrong information during the operation system installation (e.g., on the Configure Proxy screen), then the apt package manager will not work properly.

Here are some troubleshooting steps to think about:

On the SIEM VM, run the command:

```
cat /etc/apt/apt.conf.d/90curtin-aptproxy
```

This command will read the contents of the file `/etc/apt/apt.conf.d/90curtin-aptproxy` and display its contents on the screen. The file should read something like this:

```
Acquire::http::Proxy "http://172.16.1.1:3128";  
Acquire::https::Proxy "http://172.16.1.1:3128";
```

If this file does not exist, or has any content that is in any way different from the lines above, **run the following three commands exactly as displayed, and in this exact order:**

```
sudo su -  
echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-  
aptproxy  
echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >>  
/etc/apt/apt.conf.d/90curtin-aptproxy
```

This series of commands requires root access, so the first thing we do is use `sudo su -` to become the root user. The next two commands delete the current `90-curtin-aptproxy` file if it exists, then overwrites it with the two correct entries that should exist in the file. After running these commands, run `cat /etc/apt/apt.conf.d/90curtin-aptproxy` once more, and confirm that the output matches the correct output listed above. After confirming that the configuration file has been recreated correctly, try running the apt-get commands once more. If they continue to fail, then continue the troubleshooting process. Assuming that the network connectivity check commands were successful (e.g. `nslookup` and `curl`), think about the following:

- Is the SQUID proxy service installed on pfSense?
- Is there a firewall rule on the LAN interface to allow access to the proxy service? (allow traffic to IP address 172.16.1.1 port 3128 TCP from network 172.16.1.0/24)
- Is the option *Resolve DNS IPv4 First* checked on the SQUID proxy service?

These are all configurations covered in chapter 14, and should have already been specified. Double check that they have been configured correctly, then try updating the SIEM VM again.

```

ayy@siem:~$ 1 sudo su -
root@siem:~# 2 echo 'Acquire::http::Proxy "http://172.16.1.1:3128";' > /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 3 echo 'Acquire::https::Proxy "http://172.16.1.1:3128";' >> /etc/apt/apt.conf.d/90curtin-aptproxy
root@siem:~# 4 cat /etc/apt/apt.conf.d/90curtin-aptproxy
Acquire::http::Proxy "http://172.16.1.1:3128";
Acquire::https::Proxy "http://172.16.1.1:3128";
root@siem:~# 5 apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [332 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [8,780 B]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [163 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [5,404 B]
Fetched 826 kB in 1s (972 kB/s)
Reading package lists... Done
root@siem:~# 6 apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@siem:~#

```

13-71: This illustration demonstrates how to fix or modify the `/etc/apt/apt.conf.d/90curtin-aptproxy` file, in the event that student find that there is a problem with the file. First utilize `sudo su -` (1) to become the root user. Then use the two `echo` commands (2, 3) to write the correct configuration data so that `apt` knows how and where to access the squid proxy configured on the pfSense VM. Utilize the `cat` (4) command to confirm that the configuration file is properly configured. Finally, run `apt-get update` (5) and `apt-get -y dist-upgrade` (6) to check for the latest updates and download them.

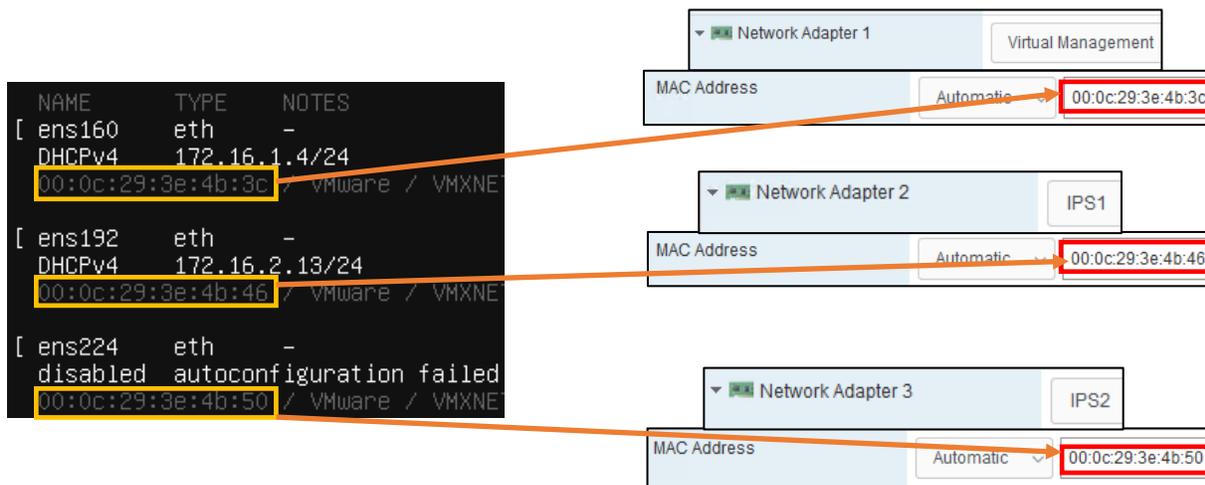
13.7.2.4 Installing Ubuntu on the IPS VM

Now that Ubuntu has been installed on the SIEM VM, network connectivity has been checked, and updates have been applied, next up is the IPS VM. The process for installing Ubuntu Server on the IPS virtual machine is practically identical, the process will be summarized below, with major differences to be aware of explained further in-depth.

- Start the IPS VM, and connect to its virtual console
- Select English as your language (or your preferred language)
- If there are any updates to Subiquity, select the option, *Update to the new installer*
- Select *English (US)* (or your preferred language) as the keyboard *Layout* and *Variant*

The *Network connections* screen will be a little bit different than it was on the SIEM VM, because the IPS virtual machine has three network interfaces. Recall in [section 13.6.3](#) (pp. 579-581), comparing and contrasting the MAC addresses of the three network adapters attached to the pfSense VM, and using that information to map the name of the network interface in pfSense (e.g., *Bridged port group* → *vmx0/WAN*, *Virtual Management port group* → *vmx1/LAN*, *IPS1 port group* → *vmx2/OPT1*).

Students will need to perform a similar exercise for the IPS virtual machine on the *Network connections* screen. In light grey text underneath the name of each network interface is the MAC address for that interface. Cross-reference the MAC address and interface name on the screen with the MAC address of adapters 1-3 recorded earlier. See *fig. 13-72* below for an example, based on the MAC addresses of my IPS virtual machine.



13-72: Ubuntu has assigned our three virtual adapters the interface names *ens160*, *ens192*, and *ens224*. Below each interface name is a MAC address. By cross referencing those MAC addresses to the MAC addresses and the virtual switches attached to the IPS VM, we can determine which interface name maps to which network adapter, and confirm which virtual switch the interfaces are attached to. For instance, *ens160*, is the network adapter attached to the *Virtual Management* port group. *ens192* is attached to the *IPS1* port group, while *ens224* is attached to the *IPS2* port group.

Now that students are aware of which interface corresponds to which network segment, the next step is ensuring that the interface connected to the *Management Virtual Switch* (e.g. the LAN network in pfSense) is the only interface that has an IP address assigned. In section 13.7.1, *fig. 13-60*, students created a static DHCP reservation for the IPS VM using the MAC address of the network adapter attached to the *Management Virtual Switch*, and assigned it the IP address 172.16.1.4. In *fig. 13-72* above, the interface `eth0` has the IP address 172.16.1.4. This confirms students created the static DHCP allocation correctly on the pfSense WebConfigurator, and that `eth0` is the interface connected to the Management Virtual Switch. If the network adapter attached to the LAN network does not have the correct IP address, there is a good chance that the static DHCP mapping for the IPS virtual machine is incorrect. Take a look at the side bar discussion, *Reservation for One*, for some troubleshooting recommendations.

Reservation for One

If you're here, that means that the network interface attached to the LAN/Management network didn't get the IP address 172.16.1.4. Similar to the *What Reservation?* Sidebar discussion for the SIEM VM, you'll want to check a few things:

- Check the MAC address of the network adapter attached to the *Management Virtual Switch*
- Visit *Services > DHCP Server* and Check the Static DHCP Mappings of the *LAN* interface, particularly, the entry for the IPS VM
- Compare the MAC address of the previous two locations with the MAC addresses presented on the *Network Connections* screen. You should already know which interface name maps to which MAC address and virtual switch. In my case this was the interface `ens160`
- Make any corrections to the static DHCP allocation, then restart the IPS VM. Make your way back to the *Network connections* screen, and confirm that the correct interface was assigned the correct IP address.

The image shows two screenshots from the pfSense web interface. The left screenshot shows the configuration for 'Network Adapter 1' under 'Virtual Management'. The 'MAC Address' field is set to 'Automatic' and displays the value '00:0c:29:3e:4b:3c'. The right screenshot shows a table of DHCP reservations with the following data:

NAME	TYPE	NOTES
[ens160	eth	-
DHCPv4	172.16.1.4/24	
00:0c:29:3e:4b:3c		/ VMwa

13-73: Just like with the SIEM VM, compare the MAC address of the network adapter attached to the *Virtual Management* port group. Compare that to the MAC address used to create a static DHCP mapping for the IPS VM on the LAN interface of the pfSense webConfigurator. If they don't match, correct the static DHCP mapping entry, then restart the IPS VM. Determine which interface was assigned the IP address 172.16.1.4.

The final step on the *Network connections* screen is to disable the remaining network interfaces. The interfaces connected to the IPS1 and IPS2 port groups (In the illustrations provided, these are the interfaces `ens192` and `ens224`) should never receive an IP address. The lab environment, and IPS software we'll be using does not require these interfaces to have IP addresses, so we want to take advantage of that. Students may have notice that the interface connected to the *IPS2* port group (`ens224`) doesn't have an IP address assigned, instead displaying the status: `disabled autoconfiguration failed`. Disregard this error message and follow the instructions below. Substitute the interface names `ens192` and `ens224` as necessary:

- Using the arrow keys, Highlight one of the other remaining interfaces. In my case, I chose to highlight `ens192`. Hit enter, and a dialogue box pops up.
 - Highlight the option *Edit IPv4*, and hit enter.
 - A new dialogue box appears titled *Edit ens192 IPv4 configuration*, with a single drop-down option highlighted, titled *IPv4 Method*. Hit enter again, and a list of choices appear. Use the arrow keys to select the option *Disabled*, and hit enter.
 - Use the arrow keys to highlight the option *Save*, and hit enter.
- Optional: Repeat the process again, only this time, Select *Edit IPv6*. By default, IPv6 should already be set to disabled, so this should not be necessary, but it is important to ensure these interfaces never receive an IPv4 or IPv6 address.
 - When finished, exit the *Edit ens192 IPv6 configuration* dialogue box.
- Repeat this process for the final interface. In my case, `ens224`. Disable the IPv4 Configuration (in my case, it was already set to *Disabled*) and confirm that the IPv6 configuration is already *Disabled*.

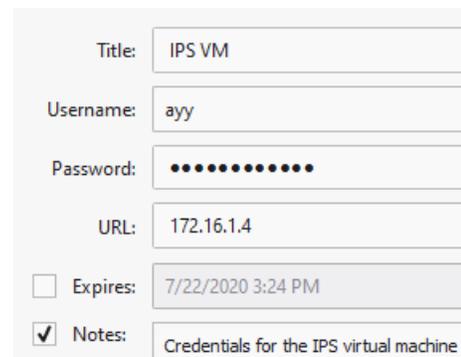
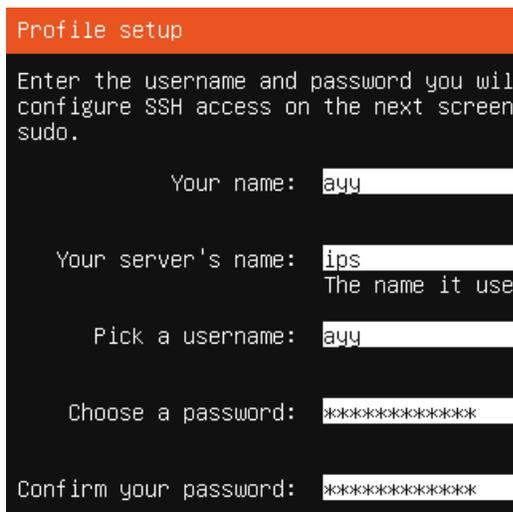
The end result should be one interface with the IP address `172.16.1.4`, and two disabled network interfaces. Students can refer to *fig. 13-74* below for assistance. When finished, use the arrow keys to highlight *Done*, and hit enter to continue.



13-74: ens160 is the interface attached to the LAN network (Virtual Management port group), and should be the only interface with an IP address. **Disable the other interfaces. They should never be assigned an IP address.**

The rest of the installation process for the IPS VM should be identical to the SIEM VM:

- On the *Configure proxy* screen, set the *Proxy address* to `http://172.16.1.1:3128`
- Accept the default archive mirror (or an alternative, if required) on the *Configure Ubuntu archive mirror* screen
- Accept the default settings on the *Guided storage configuration*, and *Storage configuration* screens. Select *Continue* on the *Confirm destructive action* dialogue pop-up
- Fill out the *Profile setup* screen, ensuring that the *Your server's name* input box is set to *ips*. Remember to document the username and password you create and store it in your preferred password manager

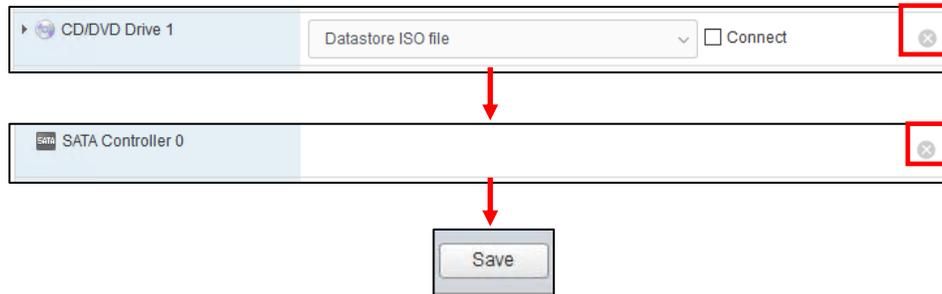


13-75: The *Profile setup* screen for the IPS virtual machine is, quite literally, the only other screen aside from the *Network connections* screen that differs from the installation process used on the SIEM VM. As always, save your username and password to a password manager!

- On the *SSH Setup* screen, be sure to select *Install OpenSSH server*
- On the *Featured Server Snaps* screen, select *Done* and hit enter to move on to the *Installation complete* phase
- Once the installation has finished, use the *Power off* option from the virtual console's *Actions* menu to shut down the virtual machine, just like the SIEM VM.

13.7.2.5 Additional Virtual Machine Settings – IPS VM

Now that Ubuntu Server is installed on the IPS VM, all that is left is to remove the *DVD/CD Drive 1*, and *SATA Controller 0* virtual hardware. The process is identical to the one used on the SIEM virtual machine – open the IPS VM's *Edit settings* menu, and click the remove icon to remove the *CD/DVD Drive 1* virtual hardware, followed by *SATA Controller 0*. As a reminder, students may have to click the *Save* button after removing *CD/DVD Drive 1*, then re-enter the menu to remove *SATA Controller 0*.



13-76: Access the IPS VM's *Edit settings* menu and remove both the *CD/DVD Drive 1* and *SATA Controller 0* virtual hardware. Remember to exit the *Edit settings* menu by clicking *Save*, then re-enter if after removing the CD/DVD drive if the SATA controller is being troublesome.

13.7.2.6 Booting the IPS VM for the first time

Power on the IPS VM, connect to its virtual console, and once Ubuntu has finished starting up and performing its first-time boot routines, log in with the username and password assigned on the *Profile setup* screen during the install. Just like with the SIEM VM, students will run the following three commands:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output from the `ip -br a` command will differ slightly, because the IPS VM has more network interfaces than the SIEM VM, but aside from that, the output from `nslookup` and `curl` should be more or less identical to the output of these commands from the SIEM VM. See fig. 13-77 below for an example on what the output of these commands should look like.

```

Ubuntu 20.04 LTS ips tty1

ips login: ayy
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)
ayy@ips:~$ ip -br a
lo                UNKNOWN      127.0.0.1/8 ::1/128
ens160            UP           172.16.1.4/24 fe80::20c:29ff:fe3e:4b3c/64
ens192            DOWN
ens224            DOWN
ayy@ips:~$ nslookup www.google.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.8.196
Name:   www.google.com
Address: 2607:f8b0:4009:815::2004
ayy@ips:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Thu, 23 Jul 2020 17:21:51 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Thu, 23 Jul 2020 17:21:51 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-23-17; expires=Sat, 22-Aug-2020 17:21:51 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=No2uEqnF70q9zD3pzs0rY1b9m4o1HDDzP4BzZ1ULDM2ia7uXqWv97cWdZN0fc2JxijI_BXyxhRfuF2EE
vFV50ssKkaJRIZPxm4TbIdfzAihP6aW6FsTqHu6Kif6j75q06iuFFU-UP0oA73r0ytPyD314nvxBKnu1_rqEmla0Sic; expires
=Fri, 22-Jan-2021 17:21:51 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-25=":443"; ma=2592000,h3-T050=":443";
ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":4
43"; ma=2592000; v="46,43"

```

13-77: Just like with the SIEM VM, students will log in to the IPS virtual machine and run a couple of network diagnostic commands. The output from curl and nslookup commands should be more or less identical to the output on the SIEM VM, but the ip -br a command will produce a few more lines of content. Ignoring the lo (loopback) interface, there should be three interfaces. Only one of them should have the status of UP. That interface should be the interface assigned to the LAN/Management network, with the IP address 172.16.1.4.

After running these commands to confirm the IPS VM has been assigned the proper IP address, can resolve hostnames, and has HTTPS connectivity, run the commands:

```

sudo su -
apt-get update
apt-get -y dist-upgrade
init 6

```

In order to become the root user, install updates on the IPS VM (and confirm the Squid proxy server is proxying the IPS VM's HTTP requests), then reboot after the system is done installing those updates.

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [306 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [114 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [7612 B]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [136 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [4792 B]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [3892 B]
Fetched 889 kB in 1s (1269 kB/s)
Reading package lists... Done
root@ips:~# apt-get -y dist-upgrade_
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for rsyslog (8.2001.0-1ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.2) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-42-generic
root@ips:~# init 6

```

13-78: These commands are identical to the ones students ran on the SIEM virtual machine, and the serve the same purpose for the IPS VM: become the root user, check for updated packages, install those updates, then reboot the system.

Note: If you're having problems with your apt-get commands failing, refer back to the sidebar conversation on pp. 614-615, [*Help! My apt-get commands are failing!*](#) For further guidance. Students can follow the exact same steps laid out for the SIEM VM to troubleshoot the problem.

13.7.2.7 Installing Kali Linux on the kali VM

Now that the SIEM and IPS virtual machines are out of the way, next up is the kali VM. Power on the VM, then Connect to its virtual console. A boot menu appears with a number of options. Using the arrow keys, highlight *Install* and hit enter.

Similar to the Ubuntu installer, the first screen, titled *Select a language*, asks users to choose the language they want to use for their installation. The default setting is *English*, use the arrow keys to highlight another language as necessary, then hit enter. The next screen, *Select your location*, asks users to choose what country, territory or area in which they are located. This screen defaults to *United States*. Use the arrow keys to change this value as necessary, and hit enter to continue. Next up is the *Configure the keyboard* screen, that asks the user what keymap to use for their installation. The default setting is *American English* and can be changed with the arrow keys. After highlighting a keymap, hit enter to continue.

The installer begins loading other phases and components it will need later. Afterwards, it will attempt to get an IP address. The pfSense DHCP server should give it an IP address through the OPT1 DHCP server, but students will not be able to confirm if the IP address 172.16.2.2 was

correctly assigned until after the operating system is installed. The next screen, titled *Configure the network*, prompts users to enter a hostname for the system. Students should use the default hostname *kali*. Hit the enter key to continue to the next screen that prompts for a domain name. Again, students may hit enter and accept the default, *localdomain*.

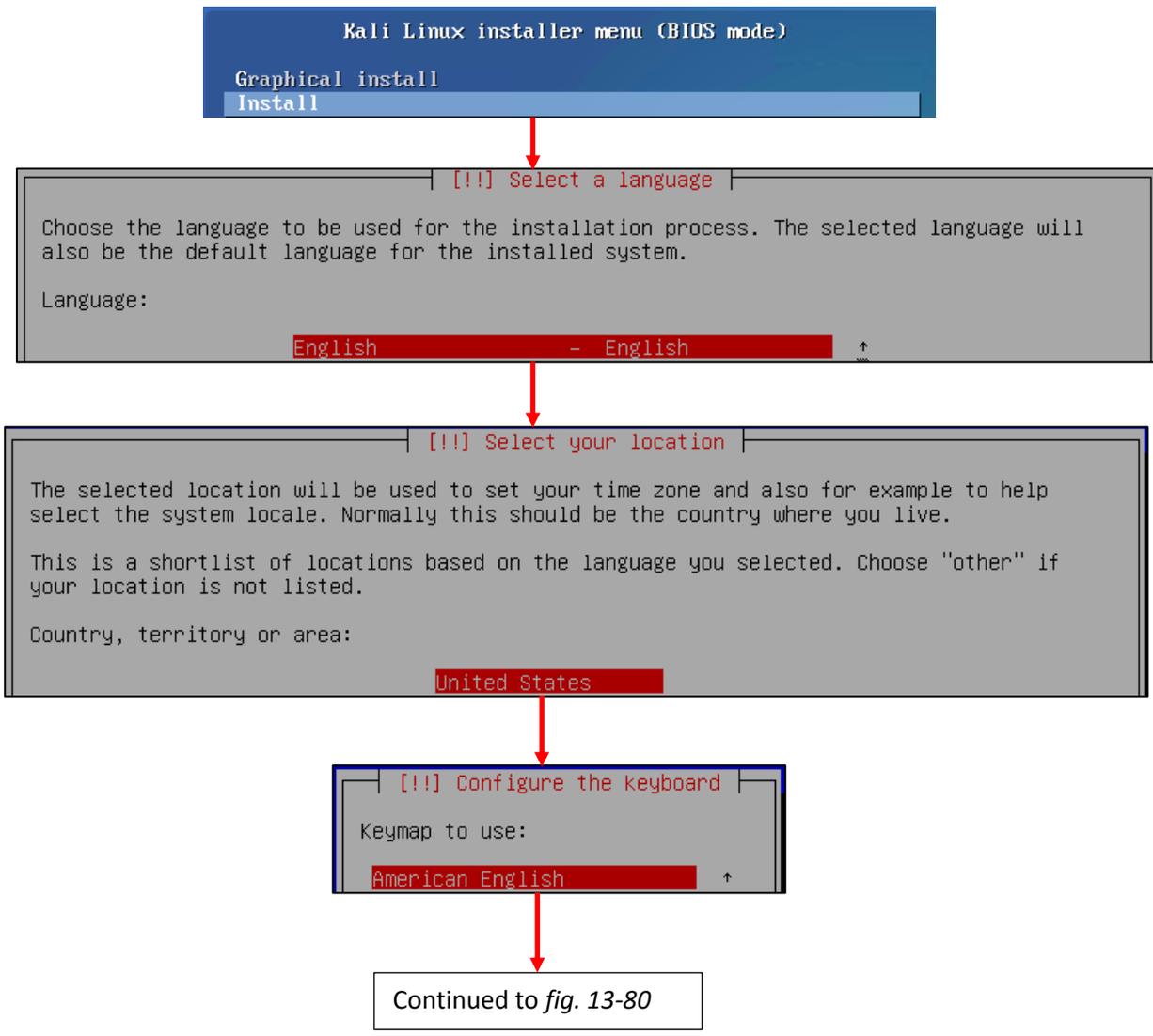
The *Set up users and passwords* screen appears. The first window asks for the full name of the user to be created. Type in the full name of the user account, and hit enter to continue to the next screen, that prompts for a username students will use to log in to the system. After typing in a username, hit enter to be prompted to create a password for this account. After hitting enter again, you'll be prompted to enter the same password again to confirm your choice. Enter the same password and hit enter to continue the installer. Just like with the SIEM and IPS virtual machines, be sure to save the username and password to your preferred password manager.

Next up is the *Configure the clock* dialogue. The installer will reach out to its preferred NTP servers to get the current time, then ask the user to select the time zone in which they are located. Use the arrow keys to choose your time zone, and hit enter to continue.

The *Partition disks* screen appears and asks users to select a partitioning method. Highlight the selection *Guided – use entire disk*, and hit enter. Users are then prompted to select the disk to partition. Since there is only a single virtual disk for the kali VM, hit enter to proceed. The next screen prompts students to select the partitioning scheme. Highlight the option *All files in one partition (recommended for new users)* and hit enter. Users are asked to confirm their choices on the next screen. Highlight the option *Finish partitioning and write changes to disk*, and hit enter. One final pop-up appears to annoy you, asking if students are sure they want to proceed, highlight *<Yes>* to confirm your choices, and press enter to continue.

The installer proceeds and begins installing the base operating systems components to the newly partitioned disk. After a moment or two, a window labeled *Software selection* appears. As the name implies, this screen allows users to pick additional software packages to install. Accept the default selections by pressing the tab key to highlight *<Continue>*, and hitting enter the next portion of the installer Retrieves and installs the requested packages. This portion of the installation may take some time, depending on internet speed and virtual machine performance.

After some time has passed a new prompt appears, labeled *Install the GRUB boot loader on a hard disk*, asking if users want to install the GRUB boot loader. This is a necessary component in order to boot the virtual machine, so highlight *<Yes>*, and press the enter key to continue. The next screen asks what partition to install the boot loader to. Seeing as how there is only one partition available, highlight it, and hit enter to proceed. After a moment or two passes, students are prompted to remove the installation media, and reboot the virtual machine to complete the installation. Just like with the SIEM and IPS virtual machines, *Turn Off* the virtual machine, then close the virtual console.



13-79: The first screens have users select their preferred language, location, and keyboard keymap.

Continued from *fig. 13-79*

[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

kali

<Go Back> <Continue>

Domain name:

localdomain

<Go Back> <Continue>

Continued to *fig. 13-81*

13-80: The next few screens configure some of the network settings. Students are prompted to enter a hostname and domain name. Students should use the default hostname of *kali*, and default domain name of *localdomain*.

Continued from *fig. 13-80*

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

ayy lmao

<Go Back> <Continue>

Username for your account:

ayy

<Go Back> <Continue>

Choose a password for the new user:

[] Show Password in Clear

<Go Back> <Continue>

Re-enter password to verify:

[] Show Password in Clear

<Go Back> <Continue>

Title:	kali VM
Username:	ayy
Password:	*****
URL:	172.16.2.2
<input type="checkbox"/> Expires:	7/24/2020 11:35 AM
<input checked="" type="checkbox"/> Notes:	credentials for the kali VM

Continued to *fig. 13-82*

13-81: Similar to the *Profile Setup* screen in the Ubuntu installer, the Kali Linux installer features a series of prompts to create a user account for the system. Be sure to save the credentials to your preferred password manager when finished.

Continued from *fig. 13-81*

[!] **Configure the clock**

If the desired time zone is not listed, then please go back to the step "Choose language" and select a country that uses the desired time zone (the country where you live or are located).

Select your time zone:

Eastern

[!!] **Partition disks**

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk

Select disk to partition:

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

Partitioning scheme:

All files in one partition (recommended for new users)

Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes

SCSI1 (0,0,0) (sda) - 85.9 GB ATA VBOX HARDDISK

#1	primary	81.6 GB	f	ext4	/
#5	logical	4.3 GB	f	swap	swap

Undo changes to partitions
Finish partitioning and write changes to disk

Write the changes to disks?

<Yes> <No>

Continued to *fig. 13-83*

13-82: After setting the time zone, students will have to configure the partitioning scheme for the install. The highlighted options above should be selected by default. If not, use the arrows to select them, and press enter to continue.

Continued from *fig. 13-82*

```
[!] Software selection

At the moment, only the core of the system is installed. The default selections below
will install Kali Linux with its standard desktop environment and the default tools.

You can customize it by choosing a different desktop environment or a different
collection of tools.

Choose software to install:

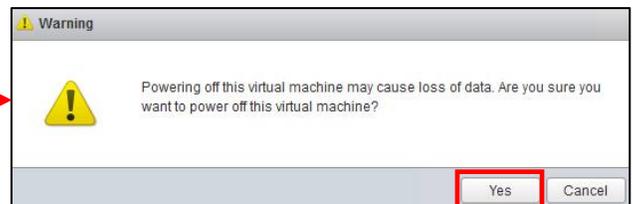
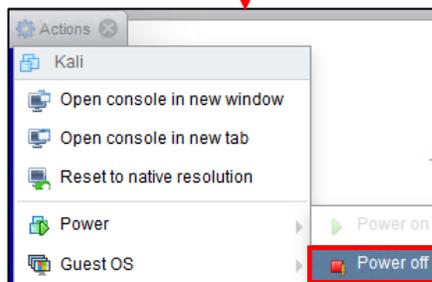
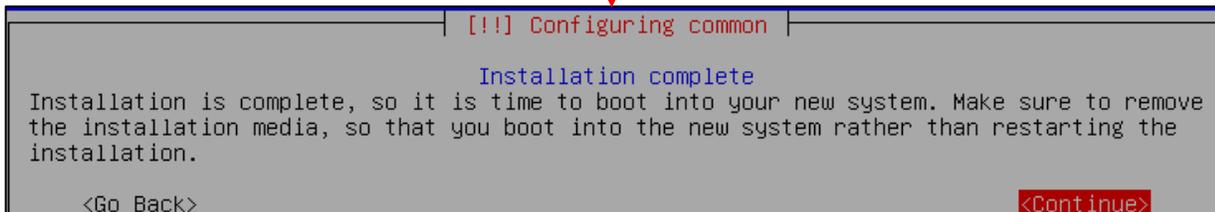
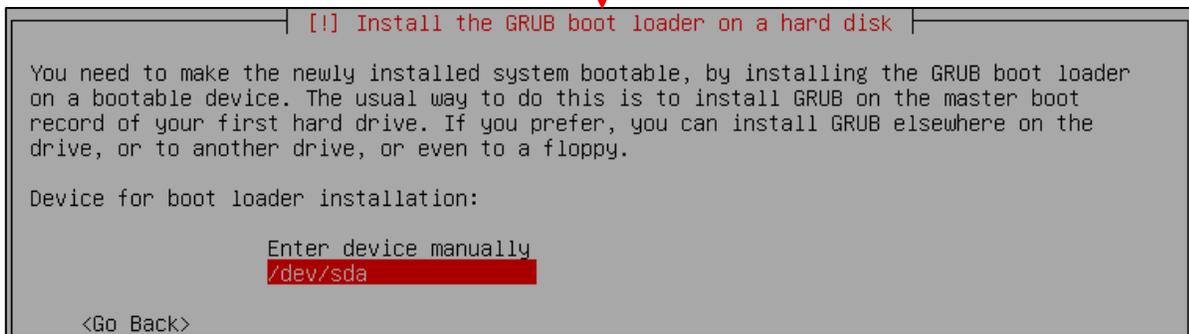
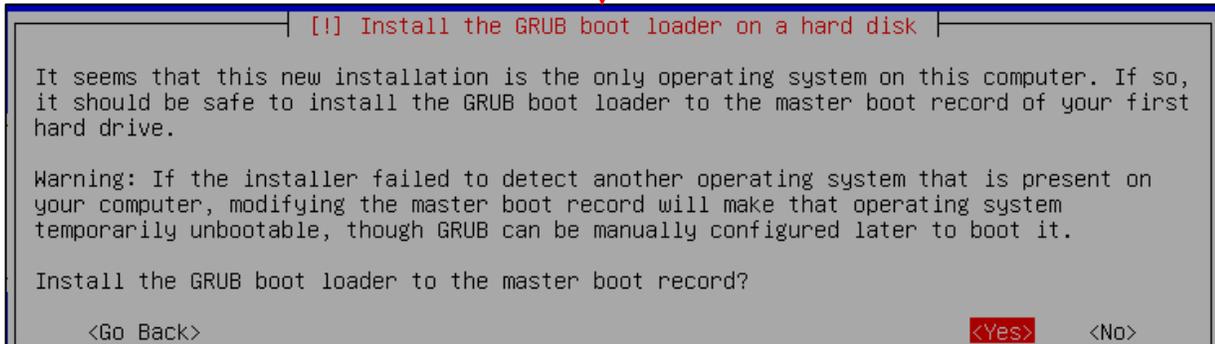
[*] Desktop environment [selecting this item has no effect]
[*] ... Xfce (Kali's default desktop environment)
[ ] ... GNOME
[ ] ... KDE Plasma
[*] Collection of tools [selecting this item has no effect]
[*] ... top10 -- the 10 most popular tools
[*] ... default -- recommended tools (available in the live system)
[ ] ... large -- default selection plus additional tools

<Continue>
```

Continued to *fig. 13-84*

13-83: On the *Software selection* screen, press the tab key to highlight *<Continue>*, and accept the default packages.

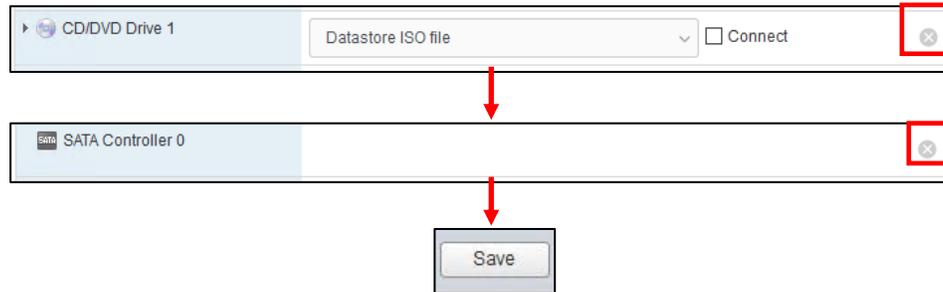
Continued from *fig. 13-83*



13-84: The final steps of the installation process. Install the GRUB boot loader to the only available disk on the system (this should be `/dev/sda`), wait for the screen labeled *Configuring common* to appear, then *Turn Off* the virtual machine and close the virtual console.

13.7.2.8 Additional Virtual Machine Settings – kali VM

By this point, it should already be established routine that once students are finished installing the operating system on their virtual machine, the next step is to remove both the *CD/DVD Drive 1* and *SATA Controller 0* virtual hardware. Kali is no exception. Access the Edit settings menu for the kali VM, remove the CD/DVD drive, followed by the SATA controller.



13-85: Access the Kali VM's *Edit settings* menu and remove both the *CD/DVD Drive 1* and *SATA Controller 0* virtual hardware. Remember to exit the *Edit settings* menu by clicking *Save*, then re-enter if after removing the CD/DVD drive if the SATA controller is being troublesome.

13.7.2.9 Booting the kali VM for the first time

With those last-minute virtual machine settings applied, *Start* the Kali virtual machine, then *Connect* to its virtual console. After a moment or two passes, students will be greeted with a graphical interface, asking for a username and password to log in. Enter the username and password supplied during the operating system install, and click *Log In* to continue.

On the top of the graphical user interface, there should be a menu bar with a few icons displayed. One of those icons is a small black window. Click on that icon to open a terminal session on the kali VM. With the terminal window open, run the same three commands that we ran on the SIEM and IPS virtual machines in order to confirm network connectivity is working as intended:

```
ip -br a
nslookup www.google.com
curl -I https://www.google.com
```

The output of `ip -br a` should confirm that only a single interface (again, ignoring the `lo` interface) is installed on the system. That interface should have the IP address 172.16.2.2. As with the SIEM and IPS virtual machines, if this is not the case, students should compare the MAC address of network adapter on the kali VM to the MAC address of the static DHCP mapping made on pfSense. **Make sure the mac addresses match, and that the mapping was created on the OPT1 interface.**

As with the SIEM and IPS VMs, `nslookup` confirms the ability of the kali VM to resolve hostnames through DNS, and the `curl` command verifies that the VM can make outbound internet

connections over HTTPS. The output of these commands should be similar to the output displayed in *fig. 13-88* below.

While Kali Linux is slightly different from Ubuntu, we can still use *most* of the same commands utilized on the SIEM and IPS virtual machines to become root, check for updates, then reboot the system. **Run these commands in this exact order:**

```
sudo su -
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
apt-get update
apt-get -y dist-upgrade
init 6
```

Students may have noticed two new commands have been added here:

```
echo 'Acquire::http::Proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
cat /etc/apt/apt.conf.d/99local
```

These commands are responsible configuring the apt package manager to use our HTTP proxy at 172.16.2.1:3128 on the *OPT1* interface of the pfSense VM. This is done by running the echo command, and redirecting its output (the > symbol) to the file /etc/apt/apt.conf.d/99local (a configuration file that the package manager will read when we run apt-get later). The second command, cat /etc/apt/apt.conf.d/99local, reads the contents of the file. If the output from the cat command reads:

```
Acquire::http::Proxy "http://172.16.2.1:3128";
```

Then that means apt was successfully configured to use the HTTP proxy. If the output from the cat command displays anything else, then students should re-enter the echo command.

Note: If most of these commands look familiar, it's because they're very similar to the troubleshooting commands I recommended in the sidebar discussion, [*Help! My apt-get commands are failing!*](#) (pp. 614-615) for the SIEM and IPS virtual machines. There are a few key differences with the kali VM to be aware of, but for the most part, the troubleshooting steps laid out are the same as the steps I laid out in this section. Here are the key differences to be aware of:

- Make absolutely sure you are redirecting the output of the echo command to the file /etc/apt/apt.conf.d/99local. ***It must be that exact file, in that exact location.***
- The kali VM doesn't need the second line, Acquire::https::Proxy "http://172.16.2.1:3128";
- Make absolutely sure to specify http://172.16.2.1:3128 as the proxy address for the kali VM.

After running these commands to configure the package manager, students should be able to run the remaining commands just like on the SIEM and IPS virtual machines. Bear in mind that Kali Linux is subject to frequent updates, and that some of those updates can be quite large. This means that depending on the performance of the Kali VM, and internet connection speeds, downloading and installing updates may take some time to complete.

ESXi: The Only Hypervisor That Tries Harder™

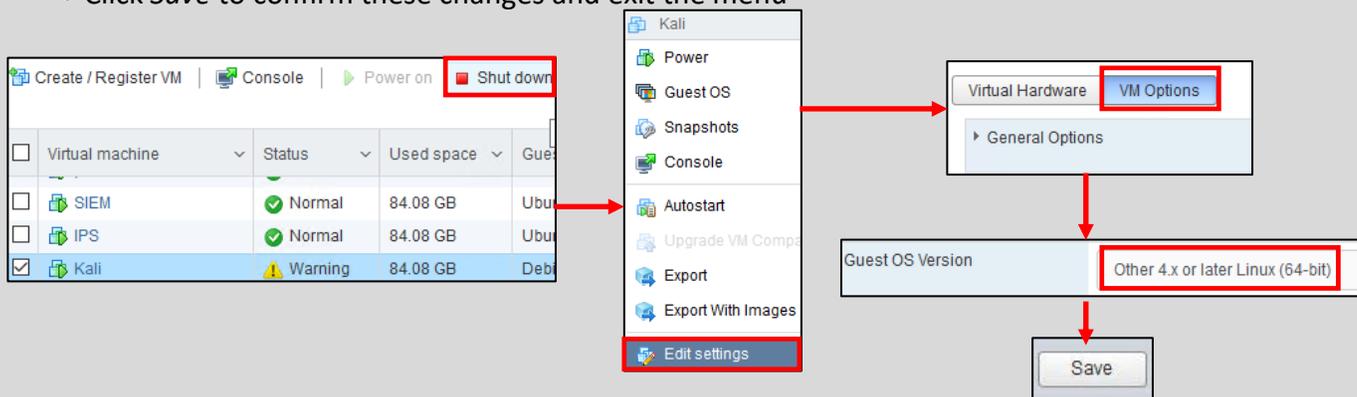
Over the course of running the kali VM, you may eventually stumble on to the configuration page for the VM that has a warning stretched across your screen:

! The configured guest OS (Debian GNU/Linux 11 (64-bit)) for this virtual machine does not match the guest that is currently running (Other 4.x or later Linux (64-bit)). You should specify the correct guest OS to allow for guest-specific optimizations. [Actions](#)

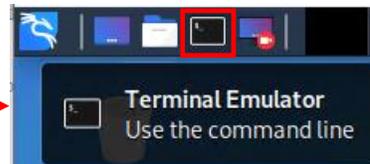
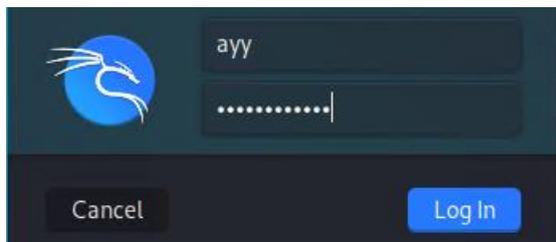
13-86: If you see this warning on the page for the kali VM, that the guest OS is misconfigured, it should be safe to ignore. Kali Linux is based on Debian. ESXi is just throwing a fit because it can't fingerprint the OS. **However...**

The warning insists that the VM isn't actually Debian Linux, but a Linux distribution that it doesn't recognize. So ESXi wants you to change the *guest OS* setting for "better performance". I assure you, Kali is based off Debian, its ESXi just failing to fingerprint the operating system due unique customizations the distro maintainer uses. In spite of this ominous warning, there should be no performance impact to the VM. However, in the event, that you are having problems with Kali Linux, and want to rule this out, here's how to fix it:

- Begin by powering off the virtual machine, using whatever means works best for you.
- Access the Kali VM's *Edit settings* menu, then click on the *VM Options* tab.
- Click the text *General Options*, and more configuration settings appear. Click the drop-down menu in the *Guest OS Version* field, and select *Other 4.x or later Linux (64-bit)*
- Click *Save* to confirm these changes and exit the menu



13-87: **...If you experience stability problems, or need to know how to do this in the future**, power off the afflicted VM, open its *Edit settings* menu, click on the *VM Options* tab and under *General Options*, change the *Guest OS Version* to the operating system ESXi recommends – in this case, *Other 4.x or later Linux (64-bit)*. Save your settings, and that's that.



```
avy@kali:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8  ::1/128
eth0              UP                172.16.2.2/24 fe80::581c:6ed2:259e:5cb/64
```

```
avy@kali:~$ nslookup www.google.com
Server:          172.16.2.1
Address:         172.16.2.1#53
```

```
Non-authoritative answer:
Name:   www.google.com
Address: 172.217.0.4
Name:   www.google.com
Address: 2607:f8b0:4009:804::2004
```

```
avy@kali:~$ curl -I https://www.google.com
HTTP/2 200
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."
date: Fri, 24 Jul 2020 19:55:47 GMT
server: gws
x-xss-protection: 0
x-frame-options: SAMEORIGIN
expires: Fri, 24 Jul 2020 19:55:47 GMT
cache-control: private
set-cookie: 1P_JAR=2020-07-24-19; expires=Sun, 23-Aug-2020 19:55:47 GMT; path=/; domain=.google.com; Secure
set-cookie: NID=204=rC1q-094PKdmAIZC2ajgCkpdGrGdulzdaxnJR2Cui-HYKBgbjzh_qCz6G5tJYoetE_Uc7rscR53x4Gri7HE3k_gu9h2BKkh6etyF0hGD0iat3FF22oe-4VngjlfAdGEY3XTtecVHB8iJ5Qw2qUVjmmN7oZGeRUSj1mXJ8ulaLkRY; expires=Sat, 23-Jan-2021 19:55:47 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: h3-29=":443"; ma=2592000,h3-27=":443"; ma=2592000,h3-T050=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
avy@kali:~$ sudo su -
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for avv:
root@kali:~# echo 'Acquire::http::proxy "http://172.16.2.1:3128";' > /etc/apt/apt.conf.d/99local
root@kali:~# cat /etc/apt/apt.conf.d/99local
Acquire::http::proxy "http://172.16.2.1:3128";
root@kali:~# apt-get update
Hit:1 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
root@kali:~# apt-get -y dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@kali:~# init 6
```

13-88: Login to the kali VM, configure the apt package manager to use the SQUID HTTP proxy on *OPT1* of the pfSense VM. Afterwards, install the latest operating system updates, then reboot the virtual machine.

13.7.3 Metasploitable 2

Metasploitable 2 is slightly different than the other lab VMs, and will require a little bit of extra work to get up and running. In the previous version of this book, I recommended the use of the *VMware vCenter Converter Standalone* application for this task. I still stand by that recommendation because it's the easiest and most user-friendly way of uploading and configuring the Metasploitable 2 VM for use on the ESXi server. However, the application is Windows only, and that might present some problems for students that utilize MacOS or Linux as their desktop operating system of choice.

The most obvious work-around would be to install a Windows VM on your management workstation, and use that to download and install the converter app, but there's another way. It won't be easy, nor will it be pretty, (nor will it be stable, or recommended) but it can be done. We'll cover both the use of the converter application, as well as this alternative method.

13.7.3.1 Acquiring the vCenter Converter Application

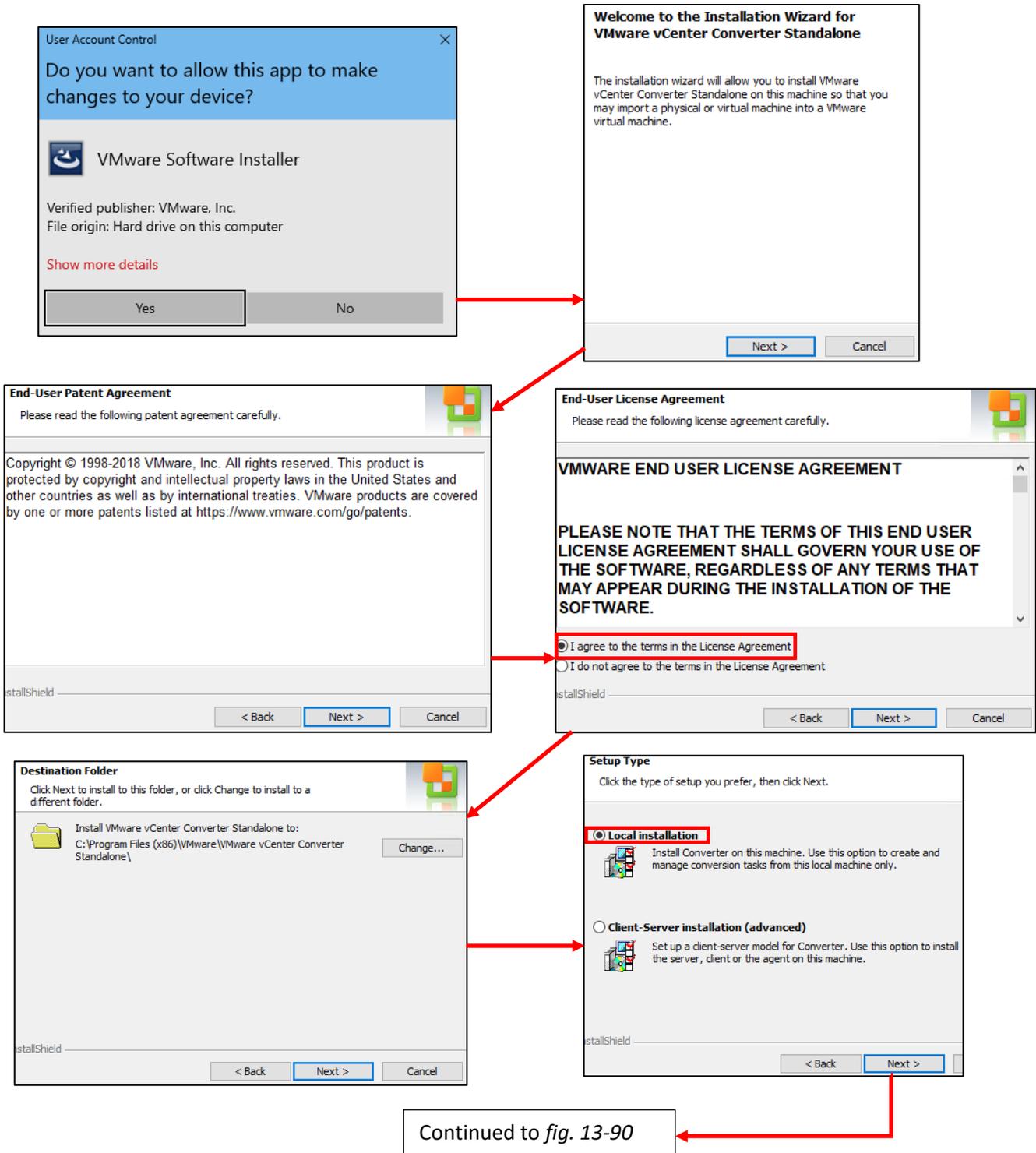
To begin, navigate to:

<https://my.vmware.com/en/web/vmware/downloads/details?downloadGroup=CONV62&productId=701&rPId=20180>

Click the large blue, *DOWNLOAD NOW* button. Students will be directed to login with their VMware customer connect account. After accepting another license agreement (that you agree to not read, as is tradition), download the installer. The current version (6.2.0.1) is 172MB.

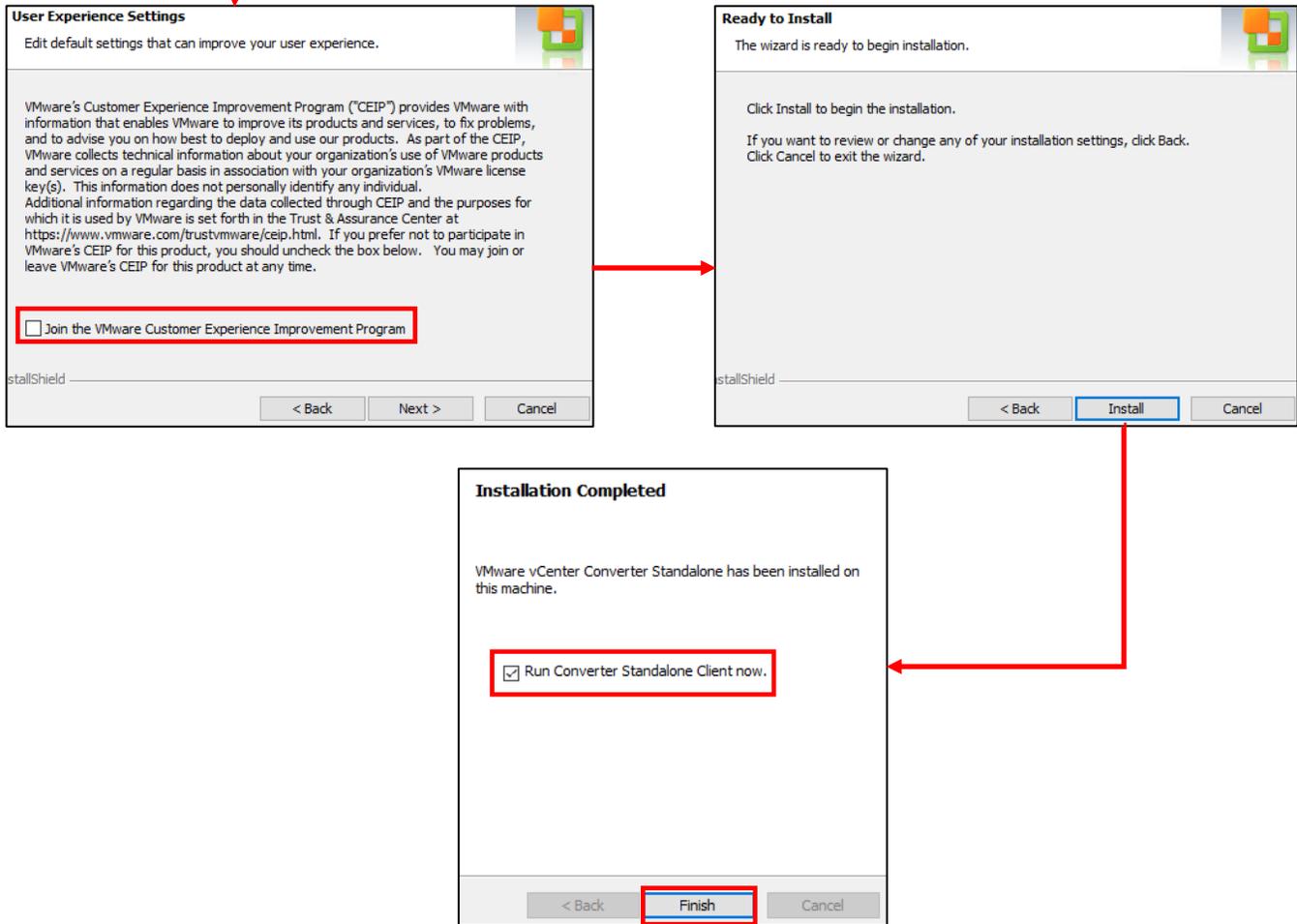
When the download completes, locate and double-click on the installer executable. If UAC is enabled, students will be prompted for permission to allow the *VMware Software Installer* to make changes to their device. Click *Yes*, and after a moment, the installation wizard will start.

Begin by clicking *Next* to proceed through the first two screens – the welcome screen and a legal disclaimer (which you agreed to not read, as is tradition). The third screen contains the terms of the end-user license agreement. Click the *I agree to the terms in the License Agreement* radio button (without reading it, because you promised), and click *Next*. The next screen allows users to change the directory the software will be installed to. Most students will know whether or not they want or need to change this – most can click the *Next* button to accept the default installation directory. The next screen, labeled *Setup Type* provides two options. Select the *Local installation* radio button, then click *Next* to proceed. The *User Experience Settings* screen allows users to opt-in (default) or opt-out (uncheck the *Join the VMware Customer Experience Improvement Program* checkbox) of sending telemetry and anonymous usage data to VMware. Make a decision, and click *Next* to continue. Finally, on the *Ready to Install* screen, click the *Install* button to proceed. After a moment or two, the *Installation Complete* screen appears, as well as a checkbox that, when checked, will start the converter application once students click *Finish*.



13-89: The installation process for the standalone converter is very straightforward. Most students will be able to safely accept the default settings with no problems.

Continued from *fig. 13-89*



13-90: The remaining steps in the installation wizard. Once again, most students can accept the default settings. Click *Finish* to close the installer and launch VMware vCenter Converter Standalone.

13.7.3.2 Converting and Uploading Metasploitable 2

Note: Before we begin, please make sure that you have downloaded the Metasploitable 2 VM from Sourceforge, and that you've decompressed the `metasploitable-linux-2.0.0.zip` file. For guidance, check out Chapter 1, [section 1.8](#) (*Using Compression Tools*, pp. 33-35). you'll need access to the files located in the `Metasploitable2-Linux` directory to perform the upload and conversion tasks ahead.

Start the VMware vCenter Converter Standalone application. On the main screen, click the *Convert machine* icon to start the VM conversion wizard. The first screen of the wizard is labeled *Source system*. Begin by selecting the *Powered off* radio button beside the text *Select source type*. In the drop-down menu immediately below the radio buttons, select *VMware Workstation or other VMware virtual machine*. In the box below, labeled *Browse for source virtual machine or image*, click the *Browse* button and browse to the `Metasploitable2-Linux` directory. Select the `Metasploitable.vmx` file, then back on the *Select source type* screen, click *Next*.

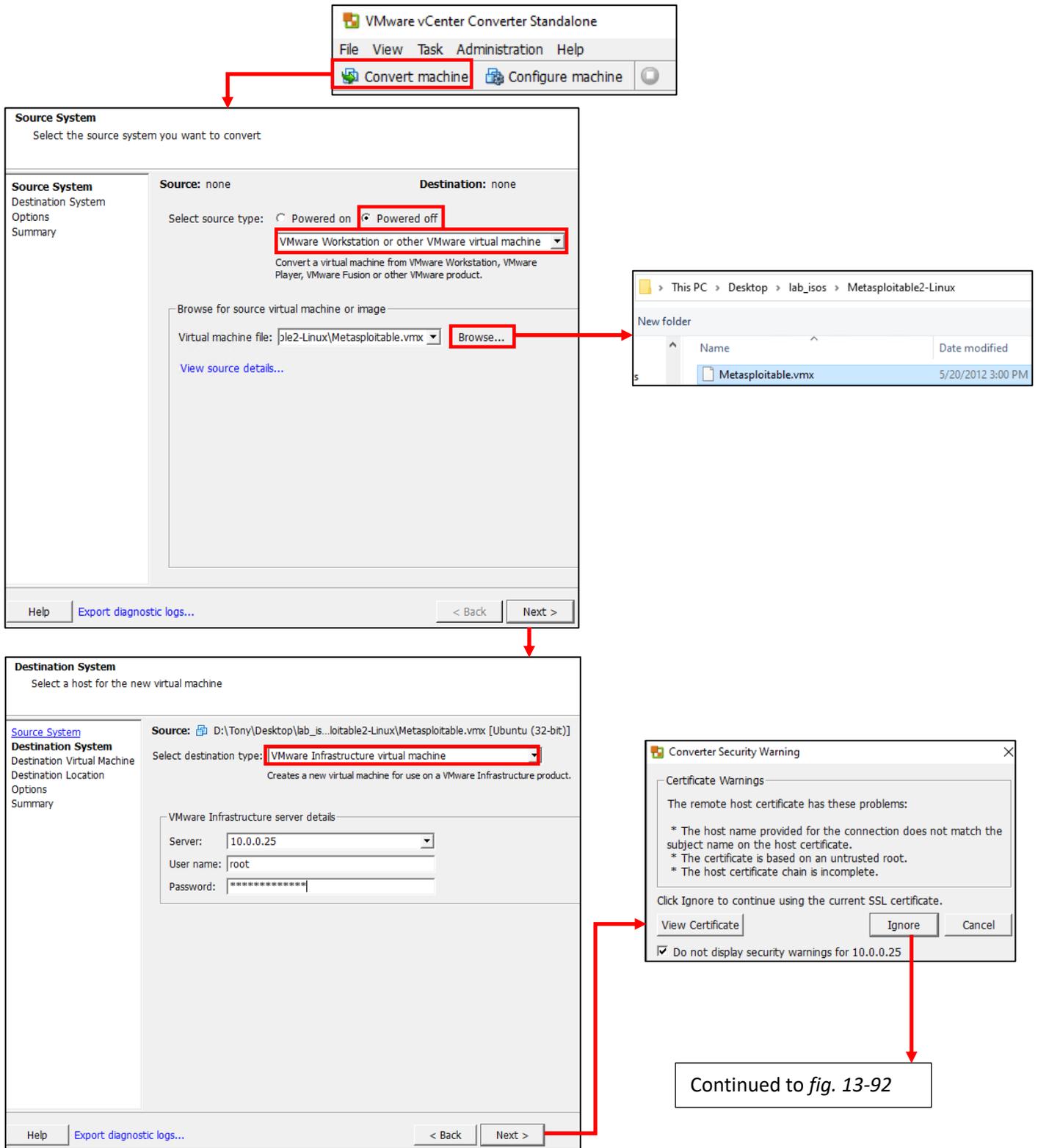
The next screen is labeled *Destination System*, in other words the wizard wants to know where to send the VM once it has been converted. In the drop-down labeled *Select destination type*, select *VMware Infrastructure virtual machine*. In the box below, labeled *VMware Infrastructure server details*, students should enter the information for their ESXi server. In the Server input box, enter the IP address used to connect to the web interface, followed by the User name root, and password for the root account. Once finished, click *Next* to proceed. Please be aware that a window titled *Converter Security Warning* will pop up letting you know that it doesn't trust the SSL certificate on your ESXi server. Assert dominance by clicking the *Do not display security warnings for [ESXi hostname/IP address]* checkbox, then click *Ignore* to proceed.

The next screen, titled *Destination Virtual Machine*, asks students what name they want to give the virtual machine at its destination. It will also display the inventory of virtual machines currently on the server. In the *Name* input box, enter *Metasploitable 2*, then click *Next*. On the *Destination Location* screen, select where the virtual machine will be stored on the ESXi server via the *Datastore* drop-down. As for the *Virtual machine version* drop-down, I recommend accepting the default configuration (which, as of this writing is *Version 17*). Click the *Next* button to proceed.

This brings students to the *Options* screen. Under the section labeled *Networks*, click the *Edit* link. The windows changes to display network adapter configuration options. Make the following changes:

- Change the *Network adapters to connect* drop-down from the default of *2*, down to just *1*
- Under the *Network* field for *NIC1*, change the port group drop-down from *Bridged* to *IPSP2*
- Under the *Controller type* field, change the drop-down from the default of *Auto* to *E1000*

After making these changes, click *Next* to proceed to the *Summary* page. After reviewing the summary for the Metasploitable 2 VM, click *Finish*, and monitor the process of the conversion.



13-91: Welcome to vCenter Converter Standalone. Click the *Convert machine* button to get started. On the *Source System* screen, select *powered off* as the *source type*, and *VMware workstation or other VMware virtual machine* from the drop-down. Click the *Browse* button and location the *Metasploitable.vmx* file. On the *Destination System* screen, select *VMware Infrastructure virtual machine* as the destination type, then enter the IP address and root user's credentials in the server details portion of the screen. The converter will warn you that the SSL certificate for the ESXi server is self-signed – Click the *Ignore* button to proceed.

Continued from *fig. 13-91*

Destination Virtual Machine

Select the destination VM name and folder

Source System: [Source System](#)
Destination System: [Destination System](#)
Destination Virtual Machine: [Destination Virtual Machine](#)

Source: [D:\Tony\Desktop...2-Linux\Metasploitable.vmx \[Ubuntu \(32-bit\)\]](#) Destination: [Metasploitable 2](#)

Name: **Metasploitable 2**

Destination Location: [Destination Location](#)
Options: [Options](#)
Summary: [Summary](#)

Inventory for: 10.0.0.25 Search for name with:

VM name	Power state
TPS	Running
Kali	Running
SIEM	Running
pfSense	Running

Refresh

Help | [Export diagnostic logs...](#) < Back Next >

Destination Location

Select the location for the new virtual machine

Source System: [Source System](#)
Destination System: [Destination System](#)
Destination Virtual Machine: [Destination Virtual Machine](#)
Destination Location: [Destination Location](#)
Options: [Options](#)
Summary: [Summary](#)

Source: [D:\Tony\Desktop...inux\Metasploitable.vmx \[Ubuntu \(32-bit\)\]](#) Destination: [Metasploitable 2](#)

Inventory for: 10.0.0.25 Total source disks size: 8 GB

ESXIServer.hsd1.mi.comcast.net

Datastore: **Mune**

Capacity: 465.5 GB
Free: 380 GB
Type: VMFS6
Block size: 1 MB

Virtual machine version: Version 17

Refresh

Help | [Export diagnostic logs...](#) < Back Next >

Continued to *fig. 13-93*

13-92: Students will need to name their virtual machine, then choose the datastore that will be used to store its files (via the *Datastore* drop-down) It is recommended to leave the *Virtual machine version* drop-down in its default state. Typically, the wizard will pick the highest version available (e.g., Version 17).

Continued from *fig. 13-92*

Options
Set up the parameters for the conversion task

Source: D:\Tony\D...ux\Metasploitable.vmx [Ubuntu (32-bit)] Destination: Metasploitable 2 on ESXiServer.hsd1.m...

Click on an option below to edit it.

Current settings:

- ▼ **Data to copy** [Edit](#)
 - Copy type: Disk-based
 - VirtualDisk1: 8 GB
- ▼ **Devices** [Edit](#)
 - vCPUs: 1 (1 sockets * 1 ...)
 - Disk controller: Preserve ...
 - Memory: 512MB
- ▼ **Networks** [Edit](#)
 - NIC1: IPS2
- ▼ **Advanced opt...** [Edit](#)
 - Power on destination: No
 - Install VMware Tools: N/A
 - Customize Guest OS: N/A
 - ⚠ Reconfigure: N/A
- ▼ **Throttling** [Edit](#)
 - CPU: None
 - Network bandwidth: None

Network adapters to connect: 1

Network adapter	Network	Controller type	Connect at power-on
NIC1	IPS2	E1000	<input checked="" type="checkbox"/>

Help | [Export diagnostic logs...](#) | < Back | Next > | Cancel

Source system information

Source type: VMware Workstation or other VMware virtual machine
 Path: D:\Tony\Desktop\lab_isos\Metasploitable2-Linux\Metasploitable.vmx
 CPU throttling: None
 Network throttling: None

Destination system information

Virtual machine name: Metasploitable 2
 Hardware version: Version 17
 Host/Server: 10.0.0.25
 Connected as: root
 VM folder: None
 Host system: ESXiServer.hsd1.mi.comcast.net
 Resource pool: Default
 Power on after conversion: No
 Number of vCPUs: 1 (1 sockets * 1 cores)
 Physical memory: 512MB
 NIC1: Connected
 IPS2
 Disk controller type: Preserve source
 Storage: Disk-based cloning
 Number of disks: 1
 Create disk 0 as: Thick provisioned disk [Mune]
 Configuration files datastore: Mune

Destination customization

Customize guest OS: No

Synchronization information

Synchronize changes that occur during cloning: No

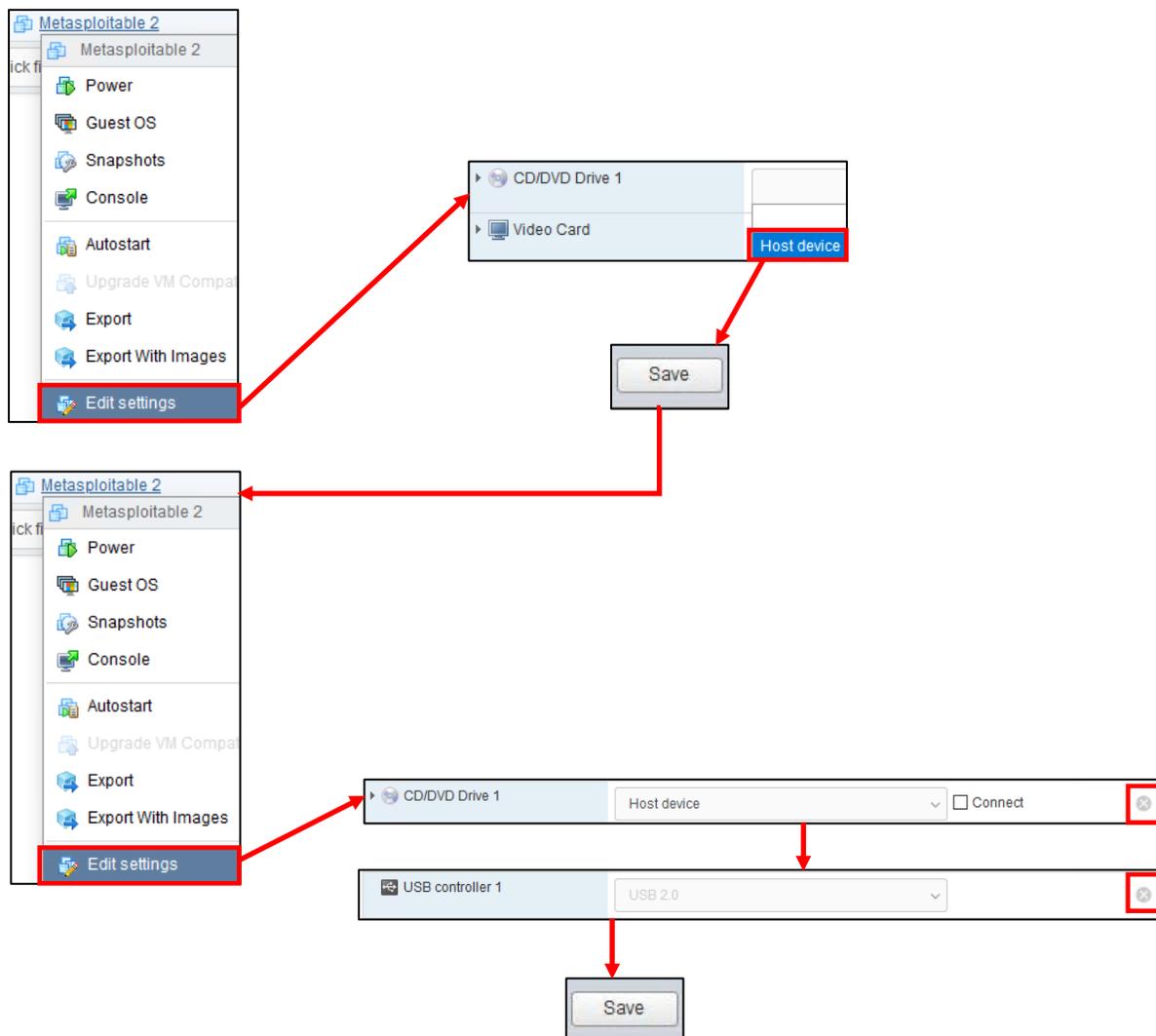
... | < Back | Finish | Cancel

Task ID	Job ID	Source	Destination	Status	Start time	End time
1	1	\Metasploitable...	10.0.0.25/Met...	✓ Completed	9/24/2020 9:1...	9/24/2020 9:15:55 PM

13-93: On the *Options* screen, click the *Edit* link to the right of the *Networks* option. Reconfigure the *Network adapters to connect* down to just 1, the *Network* setting to the *IPS2* port group, and the *Controller type* to *E1000*. Review the configuration settings on the *Summary* screen, then click *Finish* to exit the wizard, and monitor the status of the conversion task.

13.7.3.3 Additional Adjustments

The vCenter Converter will get students 99% of the way there, but there are still some slight edits to be made to Metasploitable 2. On the ESXi web interface, on the *Virtual Machines* page, right-click on the Metasploitable 2 entry, select *Edit settings*, and locate the *CD/DVD Drive 1* entry. Much like a zombie, it can't be removed without a little bit of expertise. Click the empty drop-down, then the *Host device* option. Click *Save* and exit the Edit settings menu. Afterwards, immediately right-click on the Metasploitable 2 entry in the virtual machine inventory again, and select *Edit settings* once more. This time, click the grey circle with the white X to remove the *CD/DVD Drive 1* virtual hardware like normal, as well as *USB controller 1*. Once finished, click *Save*.



13-94: Due to some inconsistencies with the vCenter converter, the *CD/DVD Drive 1* can't immediately be removed without reassigning the device it is attached to first. Students should open the *Edit settings* menu for the Metasploitable 2 VM, reassign *CD/DVD Drive 1* to the *Host device* setting, then *Save* the configuration. Immediately afterwards, re-enter the *Edit settings* menu, and remove both the *CD/DVD Drive 1* and *USB controller 1* virtual hardware, then click *Save* once more to confirm those changes.

13.7.3.4 Uploading and Converting the Metasploitable VM without vCenter Converter Standalone

Note: Before we jump into this, **this process of uploading/converting metasploitable 2 manually is more difficult than using the vCenter converter software, and is not newbie-friendly.** If you have difficulty in following these instructions, my only recommendation would be to delete any file uploaded to your ESXi server's datastores attempting this method, create a Windows virtual machine (either on ESXi itself, or your management workstation) and use that to download and run the vCenter converter application.

All that aside, just like with the converter application, make sure that you have downloaded the Metasploitable 2 VM, and that you've decompressed the `metasploitable-linux-2.0.0.zip` file. For guidance, check out Chapter 1, [section 1.8](#) (*Using Compression Tools*, pp. 33-35).

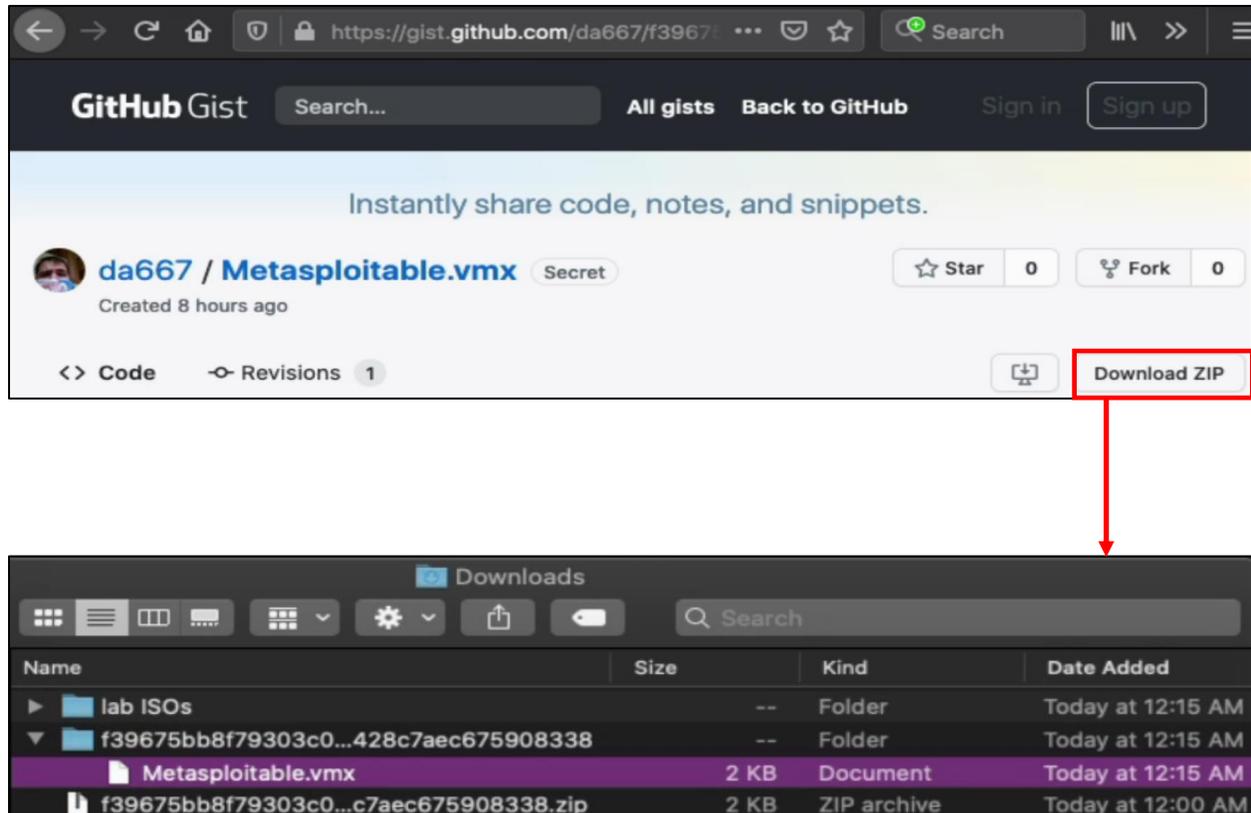
If students do not have easy access to a Windows workstation to run the VMware vCenter Converter Standalone application, all hope is not lost. **Follow these steps exactly to create the Metasploitable 2 VM.** I will be performing this demonstration using MacOS 10.15.6, but the process should be nearly identical for most Linux users.

To begin, we will be using a custom vmx file for Metasploitable 2 that strips things down as much as possible. A .vmx file is a core file used with VMware products. Think of it as a configuration file that defines what hardware is attached to the virtual machine, and the parameters for the defined hardware. Students can acquire the stripped-down vmx file through one of two methods:

Method 1: I stored a copy of the file at:

<https://gist.github.com/da667/f39675bb8f79303c06c30527e469c424>

Click the Download ZIP button. Navigate to your browser's Downloads directory, and locate the zip file. Using your operating system's compression utilities, decompress the zip file to create a folder that contains the file, Metasploitable.vmx.



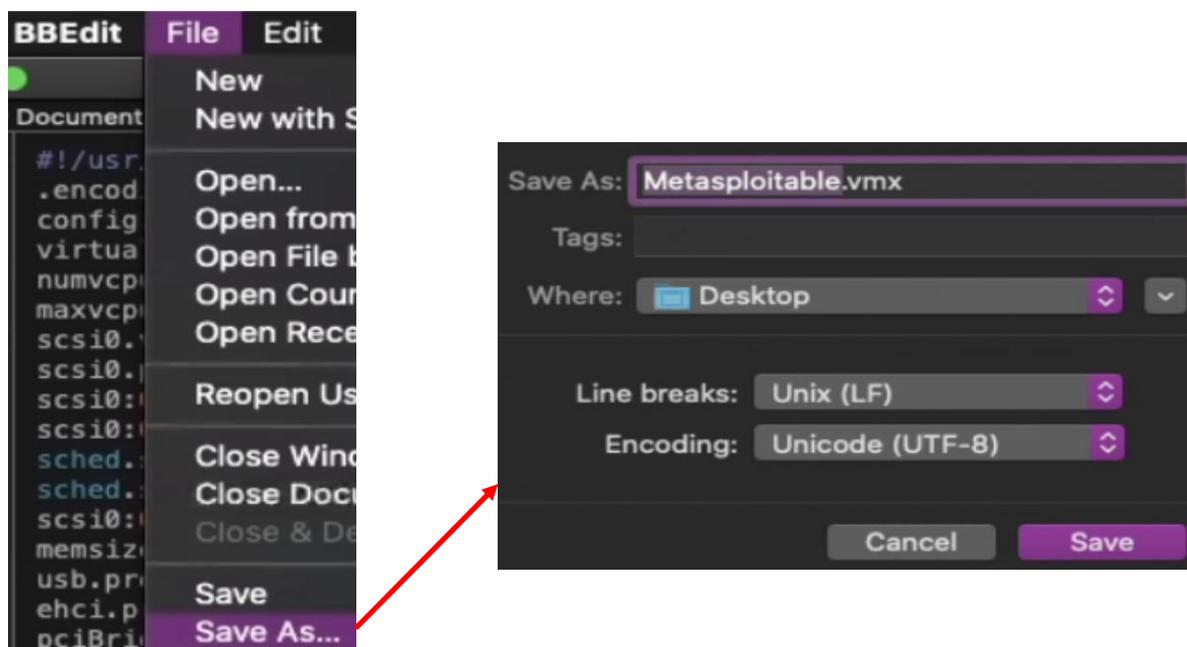
13-95: visit the github gist above and click the *Download ZIP* button. On your workstation, navigate to the current user's Downloads directory, and locate the stupidly long zip file that was just recently downloaded. Using whatever compression utilities available, unzip the zip file, and locate the `Metasploitable.vmx` file.

Method 2: In the event that students are in a region or situation where internet connectivity is spotty, here is a copy of the contents of the file from method 1

```
#!/usr/bin/vmware
.encoding = "UTF-8"
config.version = "8"
virtualHW.version = "7"
numvcpus = "1"
maxvcpus = "4"
scsi0.virtualDev = "lsilogic"
scsi0.present = "TRUE"
scsi0:0.deviceType = "scsi-hardDisk"
scsi0:0.fileName = "Metasploitable.vmdk"
sched.scsi0:0.shares = "normal"
sched.scsi0:0.throughputCap = "off"
scsi0:0.present = "TRUE"
memsize = "512"
usb.present = "TRUE"
ehci.present = "TRUE"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
vmci0.present = "TRUE"
roamingVM.exitBehavior = "go"
displayName = "Metasploitable2-Linux"
guestOS = "ubuntu"
nvram = "Metasploitable.nvram"
virtualHW.productCompatibility = "hosted"
extendedConfigFile = "Metasploitable.vmx"
tools.syncTime = "FALSE"
uuid.location = "56 4d 24 d3 2f 47 2f cd-15 1d b4 db 21 45 41 17"
uuid.bios = "56 4d 62 6f 59 bd 03 f3-e3 24 37 96 6d fa dd 2a"
cleanShutdown = "TRUE"
replay.supported = "FALSE"
replay.filename = ""
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
pciBridge7.pciSlotNumber = "24"
scsi0.pciSlotNumber = "16"
usb.pciSlotNumber = "32"
ethernet0.pciSlotNumber = "33"
ehci.pciSlotNumber = "35"
vmci0.pciSlotNumber = "36"
vmotion.checkpointFBSize = "33554432"
ethernet0.generatedAddressOffset = "0"
vmci0.id = "363079114"
tools.remindInstall = "TRUE"
checkpoint.vmState = ""
```

```
annotation = "This is Metasploitable2 (Linux)|0A|0AMetasploitable is an intentionally
vulnerable Linux virtual machine. This VM can be used to conduct security training, test
security tools, and practice common penetration testing techniques. |0A|0AThe default login
and password is msfadmin:msfadmin. |0A|0ANever expose this VM to an untrusted network (use NAT
or Host-only mode if you have any questions what that means). |0A|0ATo contact the developers,
please send email to msfdev@metasploit.com|0A|0A"
ide1:0.autodetect = "TRUE"
ide1:0.startConnected = "FALSE"
floppy0.present = "FALSE"
```

If students have a digital copy of this book, copy and paste it the text above into a text editor of some sort, and save the file as `Metasploitable.vmx` (***case sensitive***). Unfortunately, if you're using a print copy of this book, and lack internet access, you'll need to copy the content above manually into a text editor, and save it as `Metasploitable.vmx` as well.



13-96: The alternative is copy/pasting the code above out of this book (if you have a digital copy) or transcribing it manually. It's painful, but nonetheless, it is possible.

Now that students have a copy of the modified `Metasploitable.vmx` file, and the decompressed Metasploitable 2 VM, the next step is uploading everything to the ESXi server. Log on to the ESXi web interface, and under the *Navigator* pane, select *Storage*. Open the datastore browser for the datastore that will be used to hold the Metasploitable 2 VM's files. Click the *Create a new directory* button (folder with a small, green plus sign) and in the pop-up that appears, name the new folder Metasploitable 2. When finished, click the *Create directory* button. Back in the datastore browser, click on the Metasploitable 2 folder to select it, then click the Upload button (green upward pointing arrow). **Students will need to upload five files in total.**

Upload the following files from the Metasploitable2-Linux directory (the directory that gets created after decompressing `metasploitable-linux-2.0.0.zip` file):

- `Metasploitable.vmdk`
- `Metasploitable.vmx`
- `Metasploitable.nvram`
- `Metasploitable.vmsd`

With those four files uploaded, click the upload button once more, and upload the modified `Metasploitable.vmx` file that students either downloaded separately from github, or copied from this book. **All five files must be in the Metasploitable 2 directory on the datastore.** Once finished, click the Close button on the datastore browser.

13.7.3.5 Final touches

For the next step, navigate to *Virtual Machines* on the ESXi web interface, and click the *Create / Register VM* button to start the new virtual machine wizard. On the *Select creation type* screen, select the option labeled *Register an existing virtual machine*, then click *Next*. On the next screen, labeled *Select VMs for registration*, students are prompted to select a VMX file for ESXi to read, in order to register the virtual machine. Click the button labeled *Select one or more virtual machines, a datastore or a directory*, and a the datastore browser will appear. Select the datastore used to hold the Metasploitable 2 directory, and navigate to the `Metasploitable.vmx` file. Click the *Select* button to close the datastore browser, then click *Next* to proceed. Click *Finish* on the *Ready to complete* screen to add the Metasploitable 2 VM to the virtual machine inventory.

On the *Virtual Machines* pane, right click on Metasploitable2-Linux and select the *Upgrade VM Compatibility* option. A pop-up labeled *Configure VM Compatibility* appears, with a single drop-down menu labeled *Select a compatibility for the Metasploitable2-Linux upgrade*. The default selection of *ESXi 7.0 virtual machine* (or the current version of ESXi) is recommended. Click *Upgrade* to accept the defaults. A warning appears asking if students are sure they want to do this, warning about compatibility with older VMware products. Click *Yes* to complete the upgrade.

Right-click on the Metasploitable2-Linux entry in the virtual machines inventory once more. This time, select *Edit settings*. Students will need to make the following changes:

Click *Add network adapter*. Change the drop-down for the entry *New Network Adapter* from *Bridged*, to the *IPSP2* port group. Click on the text *New Network Adapter* to cause additional configuration options to appear, and on the *Adapter Type* field, change the drop-down from *VMXNET3* to *E1000*.

Click the entry labeled *Hard Disk 1* to cause additional settings related to the virtual hard disk to appear. In the field labeled *Controller location*, change the first drop-down from *SCSI controller 0* to *IDE Controller 0*

Finally, scroll down to *SCSI Controller 0* and click the grey circle with the white X to remove the virtual hardware.

When students are finished making these modifications, click *Save* to close the *Edit settings* menu.

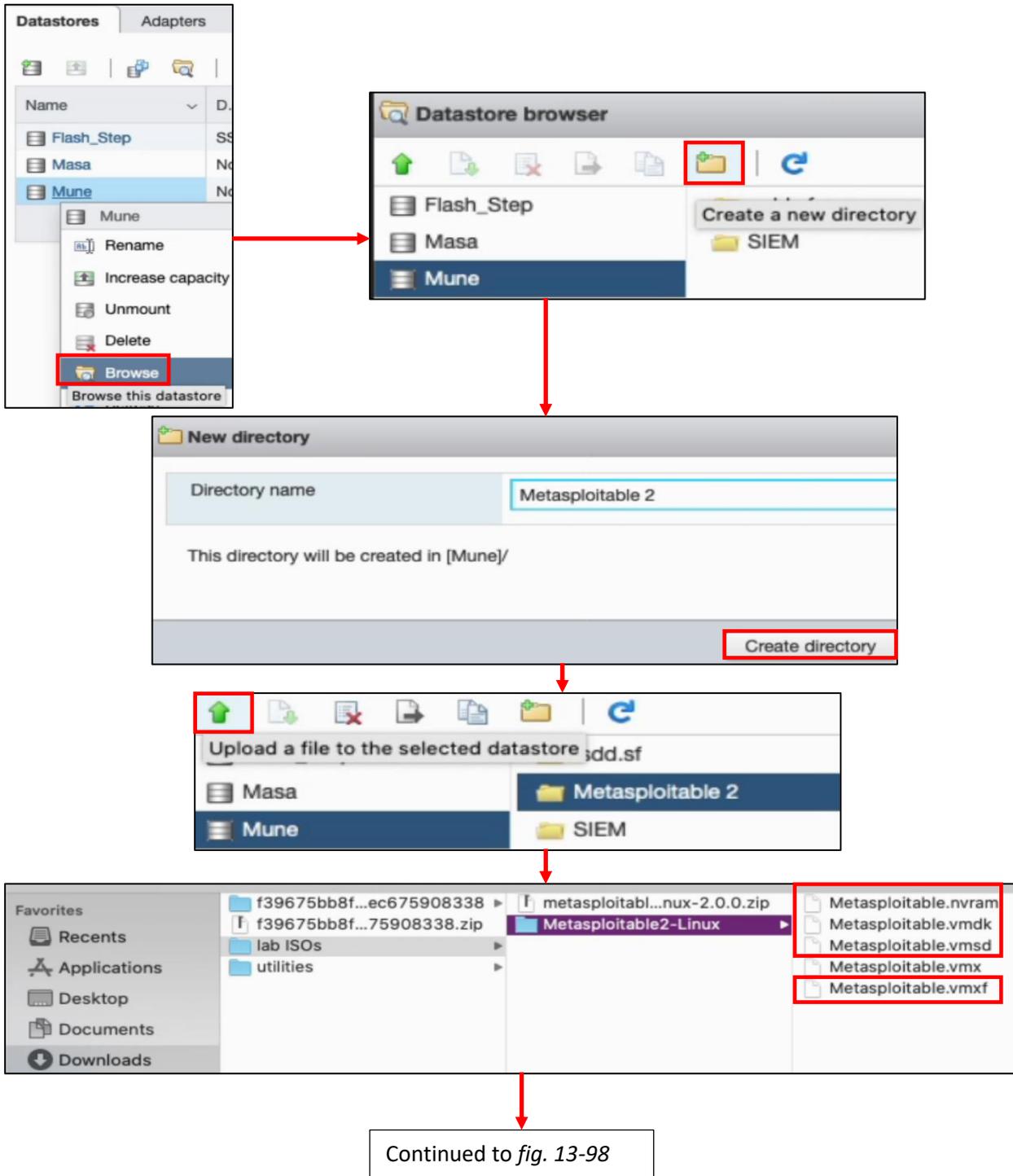
Manual Operations

As a final warning, be aware that while this "manual method" does work, and will leave you with a *mostly* functional Metasploitable 2 Virtual Machine on your ESXi server, I don't necessarily recommend it. Over the course of a few months writing this book, I noticed problems with the manually uploaded metasploitable 2 VM.

Whenever my hardware experienced power loss, the Metasploitable 2 VM would refuse to boot, and would occasionally display error messages about needing to consolidate the disks for the VM. No matter what I tried, nothing would resolve this error, except restoring the VM from a snapshot I had taken after getting it to boot. Bottom line:

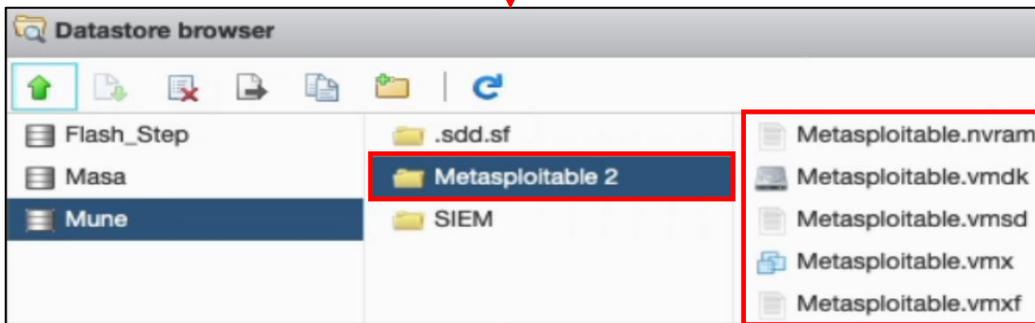
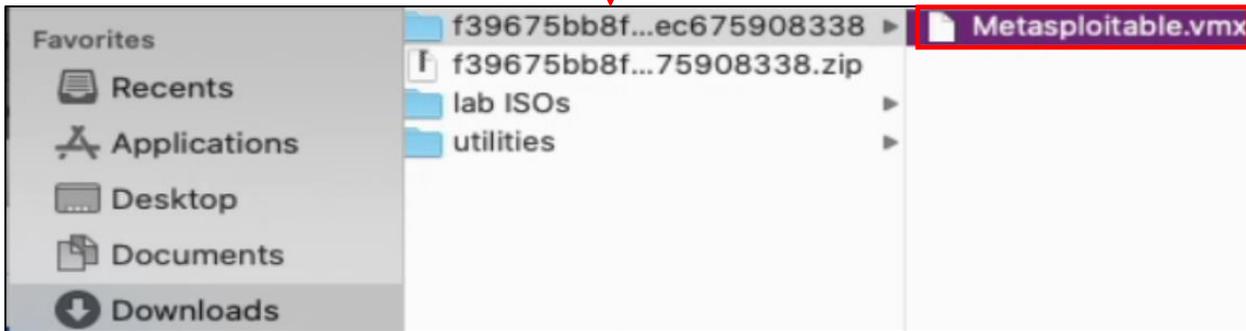
If you use the "manual method" to upload the Metasploitable 2 VM to your ESXi server, its possible you'll run into unusual error conditions in which your VM may fail to start.

Be sure to have a known good snapshot to revert back to, and if that fails to work, seriously consider using the *vCenter Converter Standalone* application to upload the Metasploitable 2 VM properly.



13-97: Navigate to Storage on the ESXi web interface, choose the datastore that will hold the Metasploitable 2 VM, then open the datastore browser. Create a folder on that datastore named Metasploitable 2. **Upload all of the files in the Metasploitable2-Linux directory except the Metasploitable.vmx file.**

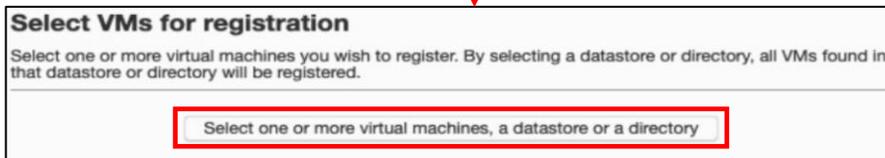
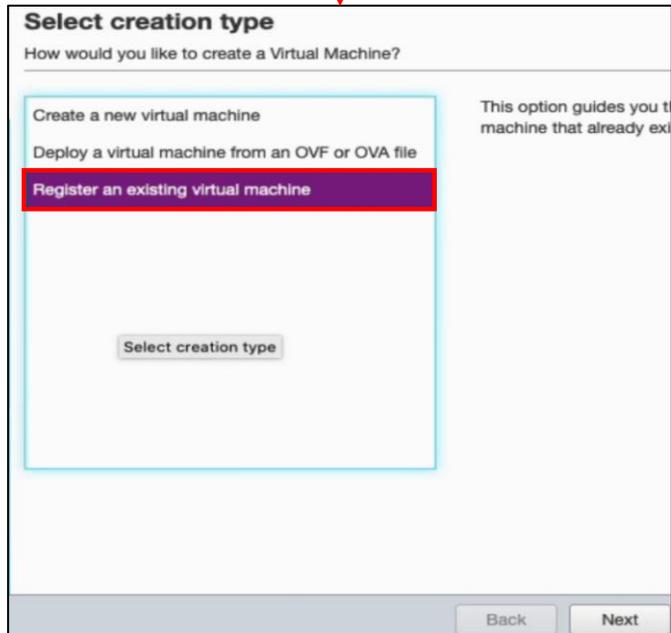
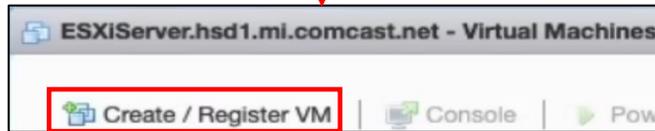
Continued from *fig. 13-97*



Continued to *fig. 13-99*

13-98: Finally, locate the modified `Metasploitable.vmx` file students are either downloaded or created separately and upload it. ***In total, students should have uploaded 5 files, and they should all be located in the Metasploitable 2 directory they created on their ESXi server.*** Click the *Close* button to exit the datastore browser.

Continued from *fig. 13-98*



Continued to *fig. 13-100*

13-99: Next up, students will need to register the new virtual machine on the ESXi server so that the hypervisor can control. To do this, click the *Create/Register VM* button on the *Virtual Machines* pane. Select the *Register an existing virtual machine* option on the *Select creation type* screen, then click *Next*. On the *Select VMs for registration* screen, click the *Select one or more virtual machines, a datastore or directory* button to open the datastore browser. Navigate to the *Metasploitable 2* directory, and select the *Metasploitable.vmx* file.

Continued from *fig. 13-99*

Select VMs for registration

Select one or more virtual machines you wish to register. By selecting a datastore or directory, all VMs found in that datastore or directory will be registered.

Select one or more virtual machines, a datastore or a directory

Remove all Remove selected

<input type="checkbox"/>	VMX file
<input type="checkbox"/>	[Mune] Metasploitable 2/Metasploitable.vmx

1 items

Back Next Finish Cancel

Ready to complete

Review your settings selection before finishing the wizard

Virtual machines	[Mune] Metasploitable 2/Metasploitable.vmx
------------------	--

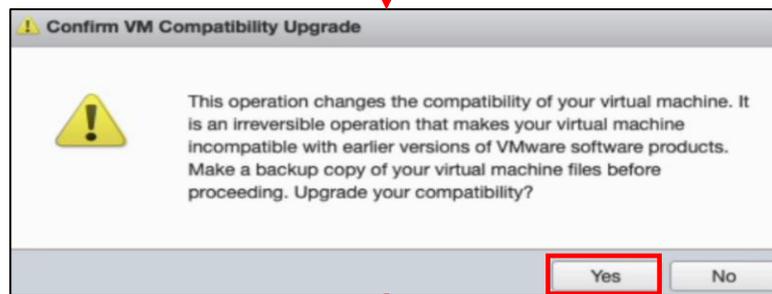
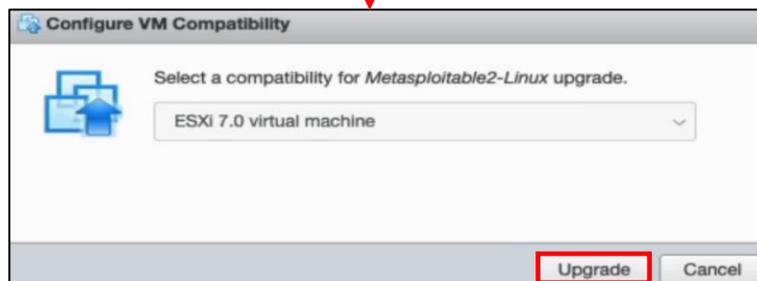
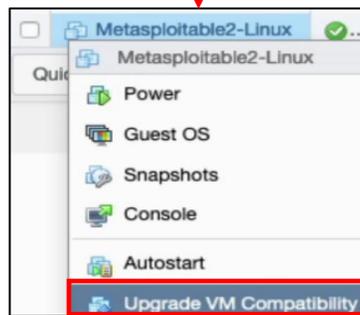
Back Next Finish

<input type="checkbox"/>	Metasploitable2-Linux
--------------------------	-----------------------

Continued to *fig. 13-101*

13-100: The table on the *Select VMs for registration* will update to display the path to the *Metasploitable.vmx* file. Click *Next* to continue to the *Ready to complete* screen, then click *Finish* to close the new virtual machine wizard. A new VM named *Metasploitable2-Linux* should appear in the virtual machine inventory list.

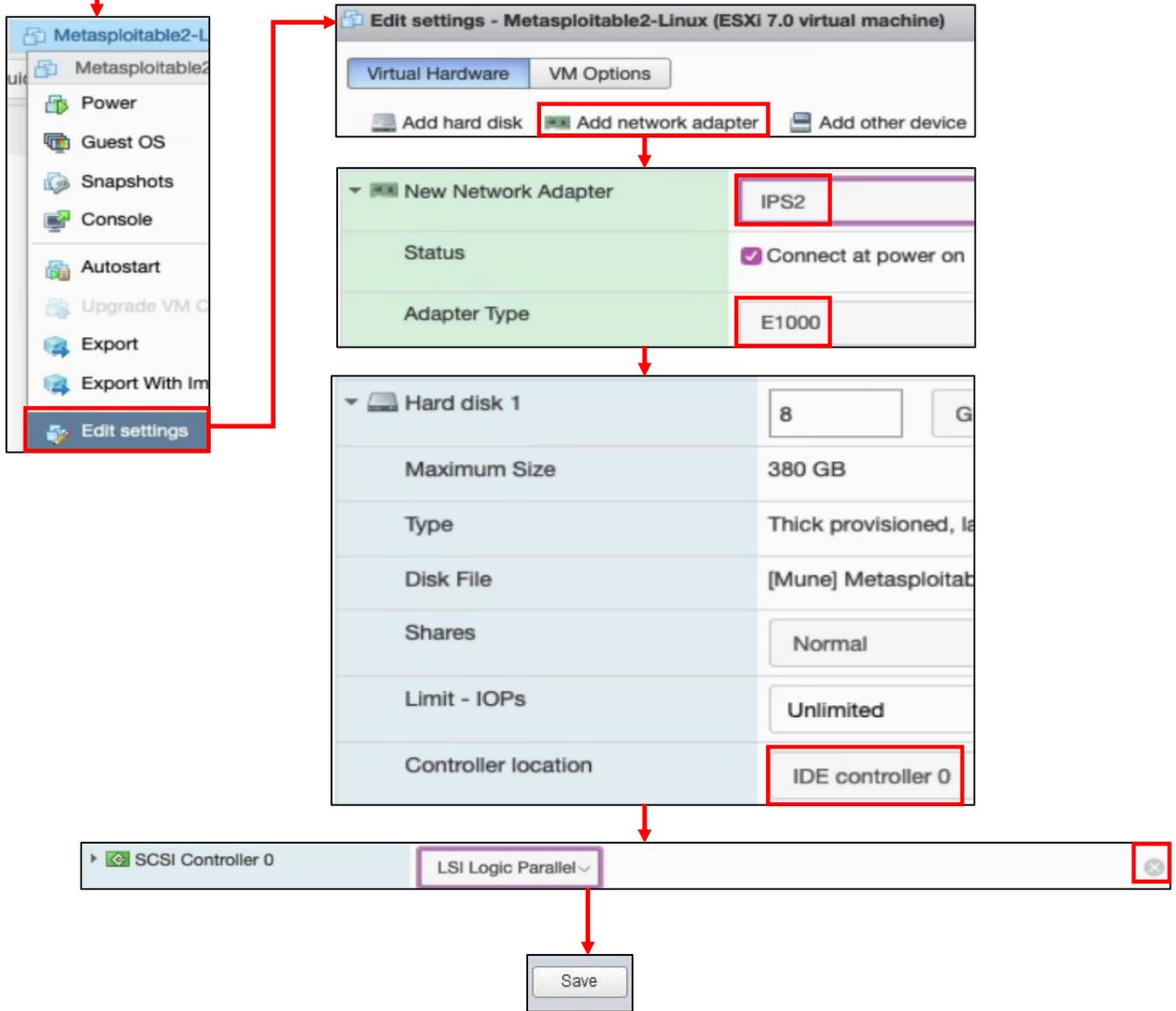
Continued from *fig. 13-100*



Continued to *fig. 13-102*

13-101: On the virtual machines inventory screen, right-click on Metasploitable2-Linux and select *Upgrade VM Compatibility*. In the pop-up that appears, accept the default setting of *ESXi 7.0 virtual machine* in the drop-down menu (or the current version of ESXi) then click *Upgrade*. A warning box pops up telling students that this will break compatibility with older versions of VMware products. Click *Yes* to complete the upgrade process.

Continued from *fig. 13-101*

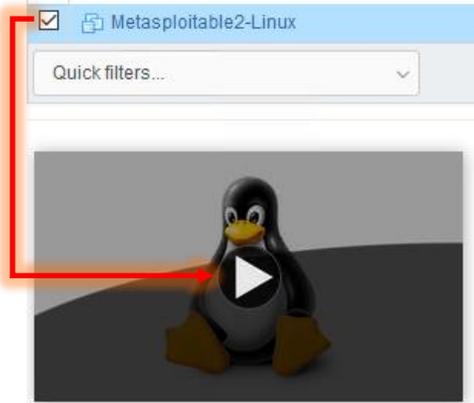


13-102: Finally, right-click on the Metasploitable2-Linux VM once more, and select *Edit settings*. Click the *Add network adapter icon*. Click on the new field that appears, *New Network Adapter* to cause additional settings related to the network interface to appear. Change the port group drop-down to *IPS2*, and the *Adapter Type* drop-down to *E1000*. Click on the *Hard disk 1* field to cause additional settings related to the virtual hard disk to appear. Find the *Controller location* field, and change the first drop-down to *IDE controller 0*. Finally, click the grey circle with a white X in the *SCSI Controller 0* field to remove it. Once finished, click *Save* to exit the *Edit settings* menu.

13.7.3.6 Metasploitable 2 Test Run

Navigate to the *Virtual Machines* pane once more (if not already there), power on the Metasploitable 2 VM, and connect to its virtual console. A whole bunch of text will scroll by as the VM goes through the boot process. After some time has passed, students should be greeted with a login prompt. The default credentials for metasploitable 2 are the username and password combination of `msfadmin/msfadmin`. Upon logging in, run the command `ifconfig -a`, and confirm that the interface `eth0` appears. **Record the contents of the field labeled `HWaddr`, the MAC address of `eth0`.** Make sure to document that it belongs to Metasploitable 2, and is connected to the IPS2 port group. When finished, type `exit` to log out of the virtual machine.

Log in to the pfSense VM's webConfigurator, and Navigate to *Services > DHCP Server > OPT1*, and create a new static DHCP mapping for Metasploitable 2, reserving the IP address 172.16.2.3.



```
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Login with msfadmin/msfadmin to get started  
metasploitable login: msfadmin  
Password: _  
msfadmin@metasploitable:~$ ifconfig -a  
eth0      Link encap:Ethernet  HWaddr 00:0c:29:47:b5:09  
          inet6 addr: fe80::20c:29ff:fe47:b509/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:2862 (2.7 KB)  
          Base address:0x2000 Memory:fd5c0000-fd5e0000  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:16436  Metric:1  
          RX packets:100 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:23481 (22.9 KB)  TX bytes:23481 (22.9 KB)  
msfadmin@metasploitable:~$ exit_
```

Services / DHCP Server / OPT1

MAC address	IP address	Hostname	Description
00:0c:29:68:d1:a4	172.16.2.2	Kali	DHCP mapping for Kali VM
00:0c:29:47:b5:09	172.16.2.3	Metasploitable2	DHCP mapping for Metasploitable 2 VM

13-103: Power on the Metasploitable VM, and connect to its virtual console. Login with the username/password of msfadmin/msfadmin. Run the command `ifconfig -a`, record the `HWaddr` field for the `eth0` interface, then type `exit` to log out. Login to the pfSense webConfigurator and add a new static DHCP allocation to the `OPT1` interface for Metasploitable 2 with the IP address 172.16.2.3.

Who Touched my VM?

If you used the manual method to upload and convert the Metasploitable 2 VM into something that resembles functional on ESXi, you'll be greeted by a pop-up that demands to know whether you moved or copied the VM before it will start up. Click the radio button labeled *I Copied It*, then click the *Answer* button to proceed.



13-104: Who touched Sasha? WHO TOUCHED MY VM?

Why aren't we doing connectivity checks?

Some of you may be wondering why we aren't doing connection checks or any of the stuff we did we for the SIEM, IPS, or Kali VMs, like checking that the static DHCP allocation is working, or attempting to connect outbound. Well, that's because right now, the metasploitable 2 VM doesn't have an IP address at all. Don't worry, its intentional, and you'll be fixing this later. The reason metasploitable 2 doesn't have an IP address is that it's connected to the *IPS2* port group. While technically the *IPS2* port group shares the same network subnet as *IPS1*, and logically it's all a part of the *OPT1* network, *IPS2* is its own physical network segment, and entirely separate from the *IPS1* network. **Without something to bridge connect the *IPS1* and *IPS2* networks together, the *IPS2* network is entirely isolated.**

Remember the network diagram back in [chapter 6](#) (p. 58)? The *IPS2* network relies on the *IPS* virtual machine being fully configured and running either Snort or Suricata in AFPACKET bridging mode. No network bridge, no network connectivity. That means no IP address from the DHCP server, either. You can see this for yourself from the output of `ifconfig -a`. You'll be fixing this later when you install either Snort or Suricata to the *IPS* virtual machine in [chapter XX](#).

```
msfadmin@metasploitable:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:47:b5:09
          inet6 addr: fe80::20c:29ff:fe47:b509/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2862 (2.7 KB)
          Base address:0x2000 Memory:fd5c0000-fd5e0000
```

13-105: `eth0` never got assigned an IPv4 address from DHCP because there is no physical connectivity between *IPS1* and *IPS2*. We'll be solving this problem later when students install Snort or Suricata to the *IPS* virtual machine.

13.8 Snapshots

The next (and final) task for students to perform will be creating baseline virtual machine snapshots for the entire lab environment. Snapshots (sometimes referred to as checkpoints by other hypervisors) instruct the virtual machine's hypervisor to gather information about the VM's current state, and save it. Later on, if there is a problem with the virtual machine such as a malware infection, or a configuration problem that cannot be diagnosed, users can choose to restore the virtual machine to its state in the past, when the snapshot was initially created

Snapshots can be created with virtual machines powered off, or while they are running, making them extremely versatile. ESXi virtual machines can also have more than one checkpoint, with the only limit being disk space required to hold them. It's extremely important to note that **virtual machine snapshots are not a substitute for backups**. If students plan on running virtual machines with important data that they cannot afford to lose, snapshots are not a substitute for backing up important files and data.

In this section, students will walk through the process of creating a virtual machine snapshot for the pfSense VM. Afterwards, it will be left as an exercise to the students to repeat the process for the SIEM, IPS, kali, and Metasploitable 2 virtual machines. Once finished, students will be ready to move on with the configuration of their lab environment.

13.8.1 How to Create a Snapshot

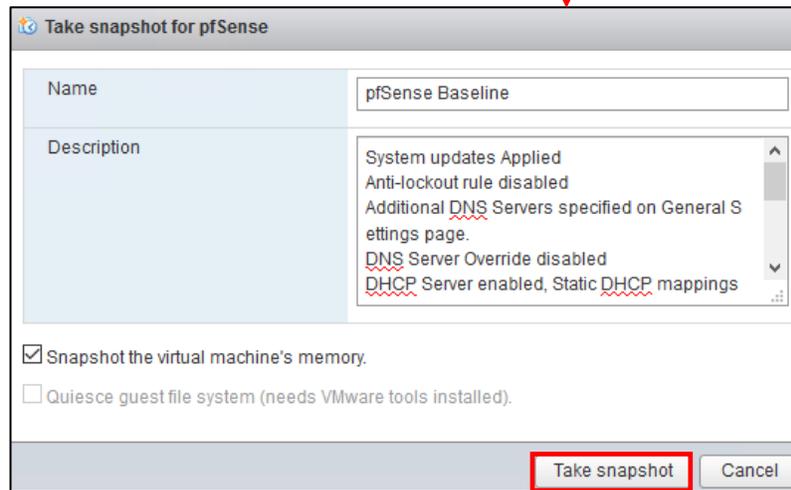
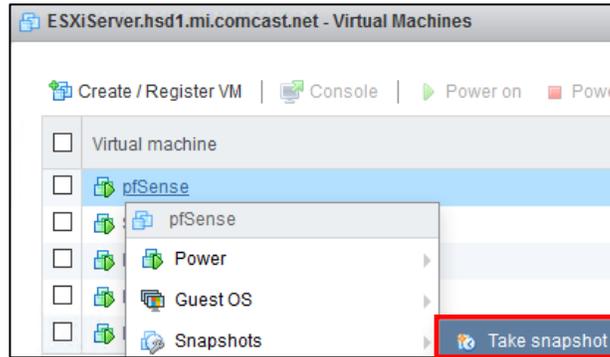
In the ESXi web interface, bring up the Virtual Machines pane, and right-click on the pfSense listing. Select the option labeled *Snapshots*, then *Take snapshot* from the sub-menu that appears. A window labeled *Take snapshot for pfSense* will appear. In the *Name* field, enter a name for this snapshot that provides a brief description about the state of the virtual machine. The *Description* field can be used to provide more detailed information about the state of the VM. For example, I recommend entering the following *Name* and *Description* for the pfSense VM's first snapshot:

Name: pfSense Baseline

Description:

System updates Applied
Anti-lockout rule disabled
Additional DNS Servers specified on General Settings page.
DNS Server Override disabled
DHCP Server enabled, Static DHCP mappings applied
DNS Resolver service enabled for LAN and OPT1
Squid proxy service installed and enabled for LAN and OPT1
NTP enabled for LAN and OPT1
Firewall policy applied for WAN, LAN and OPT1

Once finished, click the *Take snapshot* button for ESXi to begin creating the snapshot. Students can follow the progress in the *Recent Tasks* pane.



Task	Target	Initiator	Queued	Started	Result
Create Snapshot	pfSense	root	09/25/2020 18:00:07	09/25/2020 18:00:07	Completed successfully

13-106: Right-click on the pfSense entry on the virtual machine inventory pane. Select *Snapshots*, followed by *Take snapshot*. In the window that appears enter a name and description for your snapshot, then click the *Take snapshot* button. Use descriptive names, and add detailed descriptions.

13.8.2 Restoring a Snapshot

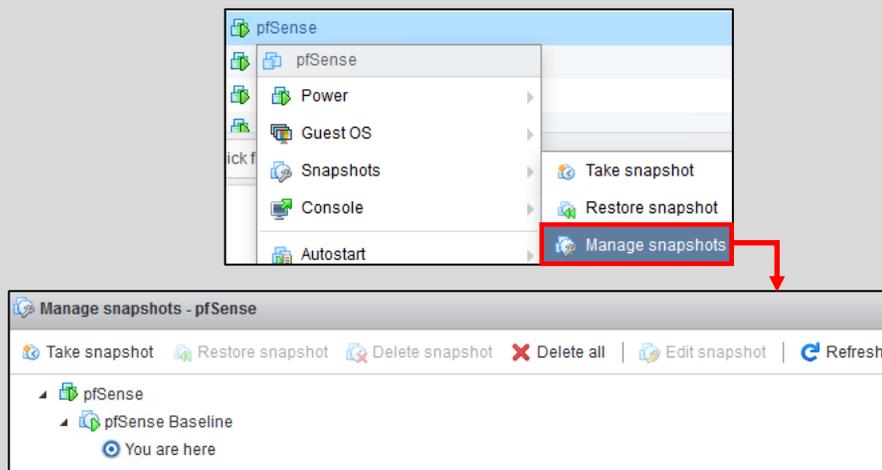
In order to restore a virtual machine snapshot, right-click the target VM in from the list on the *Virtual Machines* pane, select *Snapshots*, followed by *Restore snapshot* in the sub-menu that appears. A window appears labeled *Restore latest snapshot*. As the name of this window implies, **the Restore snapshot option only allows students to restore to the last snapshot taken on the target virtual machine.** Currently this isn't a problem, because the pfSense VM only has a single snapshot. This window warns users that restoring a snapshot means that any data on the VM between when the snapshot was made and the present time will be lost, unless it has already been backed up, or saved to another snapshot. Click the *Restore* button to continue. Students are recommended to read the sidebar conversation below *What if I have more than One Snapshot* to learn how to better manage multiple snapshots on a single virtual machine.



13-107: Restoring snapshots is even easier than making them. Right-click on the target VM, select the *Snapshots* option, followed by *Restore snapshot*. On the window that pops up, click *Restore*. After a few moments, the hypervisor will restore the virtual machine's state. Unfortunately, unlike the creation process, the restore process cannot be monitored in the *Recent tasks* pane. Have more than one snapshot? Want to restore a VM to a specific state? Want to de-clutter and remove old snapshots? Read the sidebar discussion below.

What if I have more than One Snapshot?

Clicking the *Restore snapshot* button only allows students to restore the most recent snapshot they created. That's fine if all you have is a single snapshot for your VM, but what if you have more? Right-click on the target virtual machine on the virtual machine inventory screen. Select *Snapshots*, followed by *Manage snapshots*. This will cause a new window to appear titled *Manage snapshots – [VM name]*. This is the snapshot manager. From here, you can create additional snapshots, select specific snapshots to revert to, delete snapshots you no longer have a need for, or delete every snapshot ever taken for the target virtual machine.



13-108: Access the snapshot manager for the target virtual machine, by right clicking on it on the Virtual Machines pane, selecting *Snapshots*, then *Manage snapshots*. The snapshot manager allows you to create as many snapshots as you please (so long as there is disk space available), restore to any previously taken snapshot, or delete snapshots that are no longer need. Use the power responsibly.

13.8.3 Create snapshots for the SIEM, IPS, Kali and Metasploitable 2 virtual machines

Now that students understand how to create snapshots, it is highly recommended that they create baseline snapshots for the remaining virtual machines in their lab environment – the SIEM, IPS, kali, and metasploitable 2 virtual machines. In the chapters to come, there will be a lot of complicated configuration tasks that students will need to perform in order to enable different functionality for their environment. Having a baseline snapshot to fall back to in case there are problems completing a task is handy for troubleshooting purposes.

Take snapshot for SIEM	
Name	SIEM Baseline
Description	OS installation complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied Splunk not yet installed

Take snapshot for IPS	
Name	IPS Baseline
Description	Initial OS install complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied IDS/IPS software not yet installed Splunk Forwarder not yet installed

Take snapshot for Kali	
Name	Kali Baseline
Description	Initial OS install complete Static DHCP mapping applied Network connectivity checks passed Latest updates applied

Take snapshot for Metasploitable2-Linux	
Name	Metasploitable 2 Baseline
Description	Successfully imported Confirmed VM successfully powers on Confirmed successful login Static DHCP mapping created, not yet tested No network connectivity (yet)

13-109: Now that students know how to create snapshots, apply that knowledge and make baseline snapshots for the other lab virtual machines. Having a baseline to fall back to in case something fails in the later chapters of this book is very important and will save students from a lot of headaches. ***If students received any errors while attempt to snapshot the Metasploitable 2 VM, see the sidebar conversation below, forbidden alchemy.***

Forbidden Alchemy

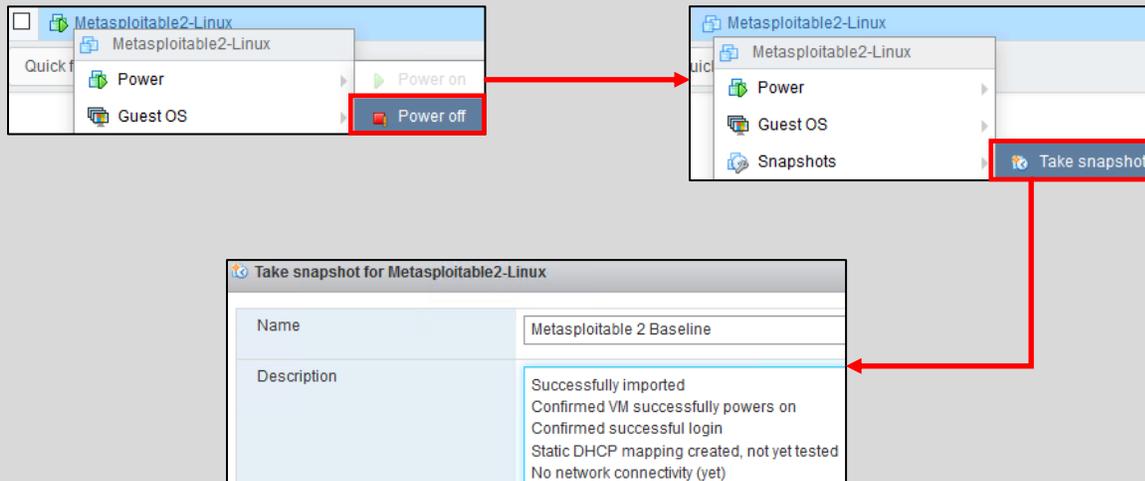
Did you get these errors when trying to snapshot the Metasploitable 2 VM?

 Failed to create snapshot 'Metasploitable 2 Baseline' on virtual machine Metasploitable2-Linux - dismiss

Create Snapshot	Metasploitable2-Linux	root	09/25/2020 22:38:02	09/25/2020 22:38:02	 Failed - Unable to access file since it is locked
-----------------	-----------------------	------	---------------------	---------------------	---

13-110: Oh. Uh. Those are errors I haven't seen before. And it coincides with me attempting a thing I haven't done before. Imagine that.

If you elected to upload the Metasploitable 2 VM using the manual method (that is, not through the vCenter Converter standalone application), this is likely an error (or some variation thereof) that you'll run into if you try to snapshot the VM while it is running. But... there is a work-around: **Power off the Metasploitable 2 VM, then create a snapshot. In fact, anything involving VM snapshots for this sin against creation that I've made should be done while the machine is powered off.** Most snapshot operations will fail if you attempt to do them while the machine is powered on. But while its powered off, its happy to add, remove and/or restore from snapshots no problem.



13-111: If you uploaded Metasploitable 2 using the "Manual method" described in section 13.7.3.4 and 13.7.3.5 (pp. 643-654), you'll need to Power off the Metasploitable2-Linux VM before taking, restoring, deleting, or otherwise performing any sort of snapshot management for the VM.

13.9 Chapter Review

Students should have all 5 virtual machines created for the baseline lab environment, as well as baseline checkpoints for all 5 virtual machines. It was a long journey to get to this point, but it's far from over. Here is a checklist of tasks to complete:

- Complete chapter 16, *Routing and Remote Access for Bare-metal Hypervisors*, starting on p. 835. In this chapter, students will learn how to enable SSH access to their lab virtual machines from Windows, Linux or MacOS through the use of a bastion host. This functionality is vital for finishing the IPS and Splunk setup guides more easily than through the VirtualBox VM console alone.
- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in chapter 17, *Network Intrusion Detection*, starting on p. 980.

Note: In section 13.5.2 (pp. 553-560), we covered configuring the virtual switches and port groups for ESXi. Specifically, allowing *promiscuous mode*, *mac addresses changes*, and *forged transmits* on the *IPS1* and *IPS2* virtual switches, combined with the *Inherit from vSwitch* setting for every port group on the ESXi server.

These configuration settings are absolutely vital to ensuring the IDS/IPS software functions correctly.

- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in chapter 18, *Setting up Splunk*, starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out chapter 19, *End of the Beginning*, starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out chapter 20, *Extra Credit*, starting on p. 1055.

Chapter 14 Patch Notes

-There is enough similarity in practically all of the hypervisor setup chapters to where initial access to the pfSense webConfigurator could be deduplicated by placing instructions on how to connect, navigating the pfSense Setup wizard, and checking for updates if they were just put into a single chapter instead. Seeing as how I've added a ton of content, deduplicating content wherever I can is valuable.

-After making sure all of the network interfaces on the pfSense VM have an IP address, and confirming internet connectivity, all readers will be forwarded to this chapter to fully configure pfSense before jumping back to their hypervisor guide chapter to finish configuring the remaining virtual machines.

-Readers are cordially invited to enable the pfSense-dark theme. Not gonna lie, I did this for me because my eyes hurt from the default theme.

-The firewall rule policies are explained on a per-interface basis, and are also divided by bare-metal vs. hosted hypervisor rule policies. We also talk quite a bit about the better anti-lockout rule, least privilege, and defense in depth as important security concepts.

-Students are provided with detailed guidance on NTP troubleshooting, choosing upstream DNS providers, DoT, DoH, DNSSEC, DNSCrypt, and how to get themselves back on the webconfigurator, if they managed to lock themselves out.

Chapter 14: pfSense Firewall Policy and Network Services

In this chapter, students will learn how to:

- Access and navigate the pfSense webConfigurator
- Complete the pfSense Setup wizard
- Check for, and install pfSense updates
- Customize the look of the webConfigurator
- Enable DNS forwarding, NTP, and Squid proxy services
- Configure static DHCP allocations
- Configure firewall aliases
- Create firewall rules
- Configure a secure firewall policy for hosted or bare-metal hypervisor labs

Be aware that some of the content (most noticeably the firewall policy) may differ depending whether or not students are using a bare-metal or hosted hypervisor.

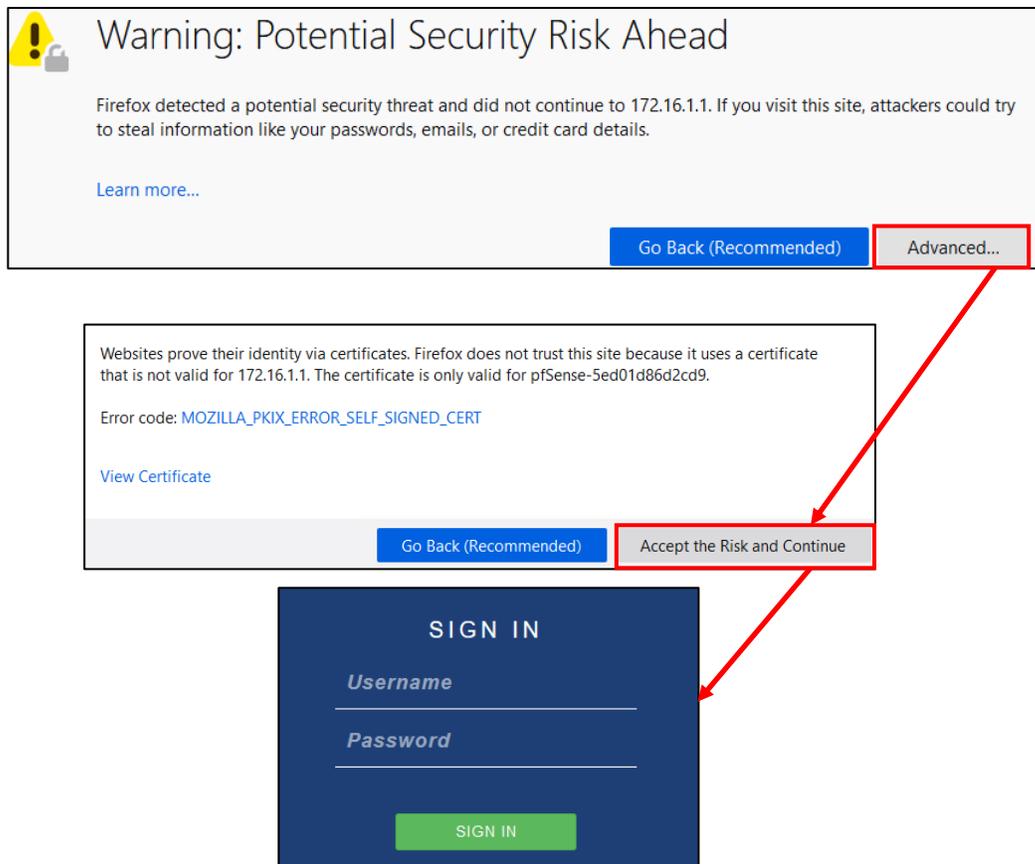
14.1 The webConfigurator, and pfSense Setup Wizard

In this section, students will learn how to access the pfSense webConfigurator, and complete the initial setup wizard. When ready, open your preferred web browser. Hosted hypervisor users (Virtualbox, VMware Workstation, VMware Fusion, and Client Hyper-V) should enter `https://172.16.1.1` in the address/URL bar.

Bare-metal hypervisor users (VMware ESXi) will need to enter the IP address of their pfSense VM's WAN interface. For example, if the IP address is 10.0.0.26, enter `https://10.0.0.26` in the browser's address/URL bar. If the web browser refuses to connect, make absolutely sure you followed the instructions provided in chapter 13 for enabling the allow all from WAN rule ([section 13.6.5.1, One Last Detail \(enableallowallWAN\)](#), p. 597). If it still refuses to connect, refer back to chapter 13 for troubleshooting guidance. Bare-metal hypervisor users should also ensure that the WAN interface of the pfSense VM is assigned a static IP address either manually (*set Interface(s) IP address* on the pfSense command-line menu), or is supplied with a static DHCP mapping from your network equipment, or IT staff.

With any luck, the web browser will present you with a screen similar to what is portrayed in *fig. 14-1* below. Most modern web browsers are extremely vocal about SSL certificates they don't trust. pfSense uses a self-signed SSL certificate to serve the webConfigurator over HTTPS. The web browser doesn't trust self-signed SSL certificates, like the one pfSense is using, and will likely throw a fit over it. Fortunately, most web browsers have buttons or dialogue options along the lines of *'yes, I accept the risk, please just let me connect.'*

For example, Mozilla Firefox presents the text, *Warning: Potential Security Risk Ahead*, convincing you that whatever site the user is trying to connect to is probably up to no good. Next to the nicely highlighted button labeled *Go back (Recommended)* is a grey button labeled *Advanced...* – click on it to open a window that explains why Firefox thinks your pfSense VM is so bad. Turns out (as mentioned above), Firefox doesn't trust the certificate pfSense is using. In this instance, we don't care what Firefox thinks is best for us, because we just set up this virtual machine and can validate that it isn't malicious. Click the button labeled *Accept the Risk and Continue* to access the webConfigurator login screen.



14-1: Web browsers are very vocal about websites they don't like or trust. Fortunately, students just created this virtual machine, and can vouch that it's probably trustworthy enough to where they don't need to care what the web browser thinks. Many web browsers have many different ways of telling users this, and they all usually have options along the lines of "I don't care, let me connect anyway." Continue connecting to access the pfSense webConfigurator login screen.

To log in to the webConfigurator, use the default username `admin`, and the password `pfSense`. After clicking the *SIGN IN* button, students will immediately be directed to the pfSense Setup wizard. This wizard is a series of screens to allow users to make a couple of basic tweaks prior to providing full control of the firewall. It can be skipped by clicking the pfSense logo on the menu bar at the top of the browser window, but it's very useful to confirm a number of configuration settings before diving in, so students are advised to follow along. The first two screens are a general welcome screen, and an additional screen to let users know Netgate offers support contracts. Click the *Next* button on these first two screens until you reach the screen labeled *Generation Information*.

General Information contains a series of input boxes, and a checkbox. The first two input boxes allow us to define a name for identifying our pfSense firewall on the network through the *Hostname* field, and a DNS suffix through the *Domain* field.

The next two input boxes, labeled *Primary DNS Server* and *Secondary DNS Server* allows users to configure IP addresses for name resolution services. If there are specific DNS servers students wish to use, they may specify up to two here. Students attached to physical networks that do not automatically provide a DNS server address to DHCP clients (or do not use DHCP at all) will want to fill out these fields with their preferred DNS server IP addresses. To put it more bluntly: If the network connectivity commands in the hypervisor chapter readers were directed from failed because pfSense couldn't resolve hostnames, this is one place to enter DNS servers to hopefully fix that problem.

Pay very close attention to this dialogue on the screen:

The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit *Services > DNS Resolver* and enable *DNS Query Forwarding* after completing the wizard.

This means ***only pfSense*** will use these DNS Servers specified for resolving domain names for its own needs (for example, resolving domains in order to download OS updates or software packages for the pfSense VM itself), but by default, pfSense will forward DNS queries from any clients (e.g., VMs in the *LAN* and *OPT1* networks) directly to the root DNS servers. The dialogue box tells users to navigate to *Services > DNS Resolver* and enable *DNS Query Forwarding* for clients to benefit from specifying DNS servers.

That leaves the *Override DNS* checkbox, that is checked by default. The text below it reads *Allow DNS servers to be overridden by DHCP/PPP on WAN*. If your local network (e.g., the bridged network/WAN interface) provides DHCP, and that DHCP lease includes DNS servers, *and* this checkbox is checked, pfSense will always use those DNS servers over any of the configured DNS servers in the *Primary/Secondary DNS Server* input boxes above. Some students may be connected to service providers or enterprise networks where they are required to use a specific set of DNS servers. If so, leave this checkbox enabled. If you prefer to use specific DNS servers, fill out the input boxes above this checkbox, and uncheck it.

Unless students have very specific use cases or unusual network requirements (you know your local network better than I do!), I recommend leaving all of the settings on this page set to their default configuration. If you are interested in setting up DNS query forwarding, and multiple upstream DNS servers, we will cover that later, in this chapter. For now, click *Next* to continue.

The next screen is labeled *Time Server Information*, and contains two fields. The first is an input box labeled *Time server hostname*. This is an NTP (Network Time Protocol) server pfSense will use for synchronizing time with. Unless users live in a specific region or their network utilizes internal NTP server(s), do not modify this field. The next field is a drop-down menu allowing users

to select a timezone. Again, unless there is a specific need, it is recommended to accept the default of UTC. Click *Next* to move to the next screen.

Users are greeted with the *Configure WAN Interface* screen. While there are a lot of settings here, and it looks pretty intimidating, students already completed the hard part through the pfSense command-line menu. Either the WAN interface received an IP address through DHCP from the hypervisor host's physical network, or the *Set interface(s) IP address* wizard was used to provide the WAN interface with a static IP address, subnet mask and upstream (default) gateway. The only settings readers need to concern themselves with on this page are towards the very bottom. Scroll down the sections labeled *RFC1918 Networks* and *Block bogon networks*. Uncheck both of these boxes. **This is extremely important for the bare-metal hypervisor users.** These are built-in pfSense firewall rules that block network requests inbound from RFC1918, and other reserved networks.

Why are these rules being disabled? Later in this chapter, readers will learn about default deny rules, and get to see the firewall policy they will be configuring for the pfSense VM, so hold out till then. After unchecking both checkboxes, click *Next*.

The next screen, labeled *Configure LAN interface*, has two fields that should already be filled out. The *LAN IP Address* should be 172.16.1.1, and the *Subnet Mask* should already be set to 24. If that is not the case, make the necessary changes, and click *Next*.

Note: Remember that if your home, school, or work network already uses the 172.16.1.0/24, or 172.16.2.0/24 networks to substitute as necessary. Each of the hypervisor setup chapters should have had a sidebar conversation labeled, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?* Check out that sidebar conversation for alternate network configurations that may better suit your needs.

The next screen is *Set Admin WebGUI Password*. Choose a suitable password and input it into the *Admin Password*, and *Admin Password AGAIN* input boxes. If readers are utilizing a password manager, now is a very good time to record the password or passphrase for the pfSense webConfigurator. Students will need to log in and make frequent changes, so it pays to have these credentials on-hand and in their password manager. Once a new admin password has been entered twice and recorded for safe keeping, click *Next* to continue.

The next screen, labeled *Reload configuration*, has a single button labeled *Reload*. Click it to progress the *pfSense Setup* wizard. The wizard will progress automatically to step 9, labeled *Wizard completed*. Users may Click the blue *Finish* button at the bottom to continue to the webConfigurator Dashboard, or click *Check for updates* to be directed to the *System Update* page.

General Information

On this screen the general pfSense parameters will be set.

Hostname
EXAMPLE: myserver

Domain
EXAMPLE: mydomain.com

The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit Services > DNS Resolver and enable DNS Query Forwarding after completing the wizard.

Primary DNS Server

Secondary DNS Server

Override DNS
Allow DNS servers to be overridden by DHCP/PPP on WAN

[» Next](#)

Time Server Information

Please enter the time, date and time zone.

Time server hostname
Enter the hostname (FQDN) of the time server.

Timezone

[» Next](#)

RFC1918 Networks

Block RFC1918 Private Networks Block private networks from entering via WAN
When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

Block bogon networks

Block bogon networks Block non-Internet routed networks from entering via WAN
When set, this option blocks traffic from IP addresses that are reserved (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and obviously should not appear as the source address in any packets received.

[» Next](#)

Configure LAN Interface

On this screen the Local Area Network information will be configured.

LAN IP Address
Type dhcp if this interface uses DHCP to obtain its IP address.

Subnet Mask

[» Next](#)

Continued to *fig. 14-3*

14-2: The *pfSense Setup* wizard, steps 2 through 5. Students should be able to leave most settings default, unless your local network has special requirements. Again, most students will know their local networks better than I will. On the *Configure WAN Interface* screen, ensure the *Block RFC1918 Private Networks* and the *Block bogon networks* checkboxes are **unchecked**.

Continued from *fig. 14-2*

Set Admin WebGUI Password

On this screen the admin password will be set, which is used to access the

Admin Password

Admin Password AGAIN

[» Next](#)

Reload configuration

Click 'Reload' to reload pfSense with new changes.

[» Reload](#)

Wizard completed.

Congratulations! pfSense is now configured.

We recommend that you check to see if there are any software updates available. Keeping your software up to date is one of the most important things you can do to maintain the security of your network.

[Check for updates](#)

14-3: The remaining steps of the pfSense Setup wizard. Set a strong password for the webConfigurator, store it in a password manager, then click *Check for updates* once the wizard is completed. Performing system updates will be covered in the next section.

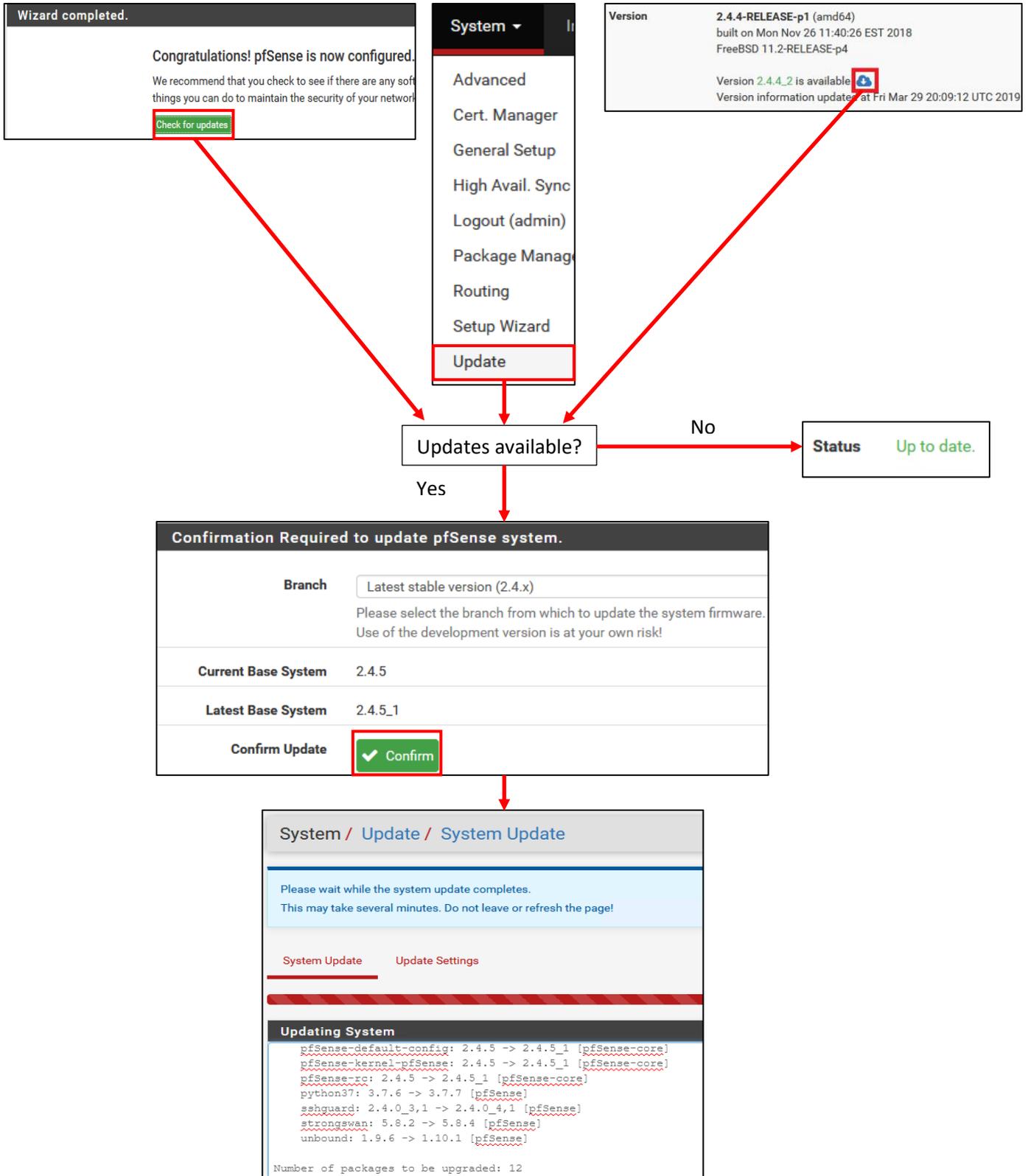
14.2 Checking for System Updates

Because a good virtual lab requires good housekeeping, it is recommended to regularly check for system updates. If students didn't already click the *Check for updates* button at the end of the *pfSense Setup* wizard, Use the pfSense navigation menu at the top of the browser window and select *System*, then *Updates* from the drop-down. Alternatively, the pfSense dashboard has a field labeled *Version* under the *System Information* window that automatically checks for system updates and notifies users of any new updates available. If there are updates available, usually a small cloud icon with a downwards facing arrow appears that can be clicked to go directly to the *System Update* page.

Note: When I first wrote this chapter, the current pfSense release was version 2.4.5_1. Coming back to edit this chapter, the current release is 2.5.1. This section demonstrates the upgrade process from 2.4.5 to 2.4.5_1. Doubtless by the time anyone is actually reading this, there will likely be more updates, but the process *should* remain mostly the same.

Once on the *System Update* page, a window will appear labeled *Confirmation Required to update pfSense system*. The drop-down menu labeled *Branch* should be set to *Latest stable version*, while the fields below display the *Current Base System*, or version of pfSense installed and the *Latest Base System*, or latest stable update available for download. The bottom most field changes depending on whether or not updates are available. If there are no updates available, the field is named *Status* and will display *Up to date*. If there *are* updates available, the field name will change to *Confirm Update*, and a green icon with a white checkbox, labeled *Confirm* appears for users to click to begin the upgrade process.

A new page with a progress bar will appear for users to watch the upgrade. Be aware that starting the upgrade process will result in the pfSense VM rebooting, and will require users to log back in once it has completed. pfSense will start a countdown that will automatically refresh the page every so often, until the login page can be reached. Once you log back in, the *Dashboard* page will be displayed and students can check the *Version* field under *System Information* to confirm pfSense is fully up to date.



14-4: The system update process for pfSense. It's very important to maintain an updated lab environment. Fortunately, the *Version* field under *System Information* on the *Dashboard* page informs users of available updates upon login.

The *System Update* Page Can be Misleading

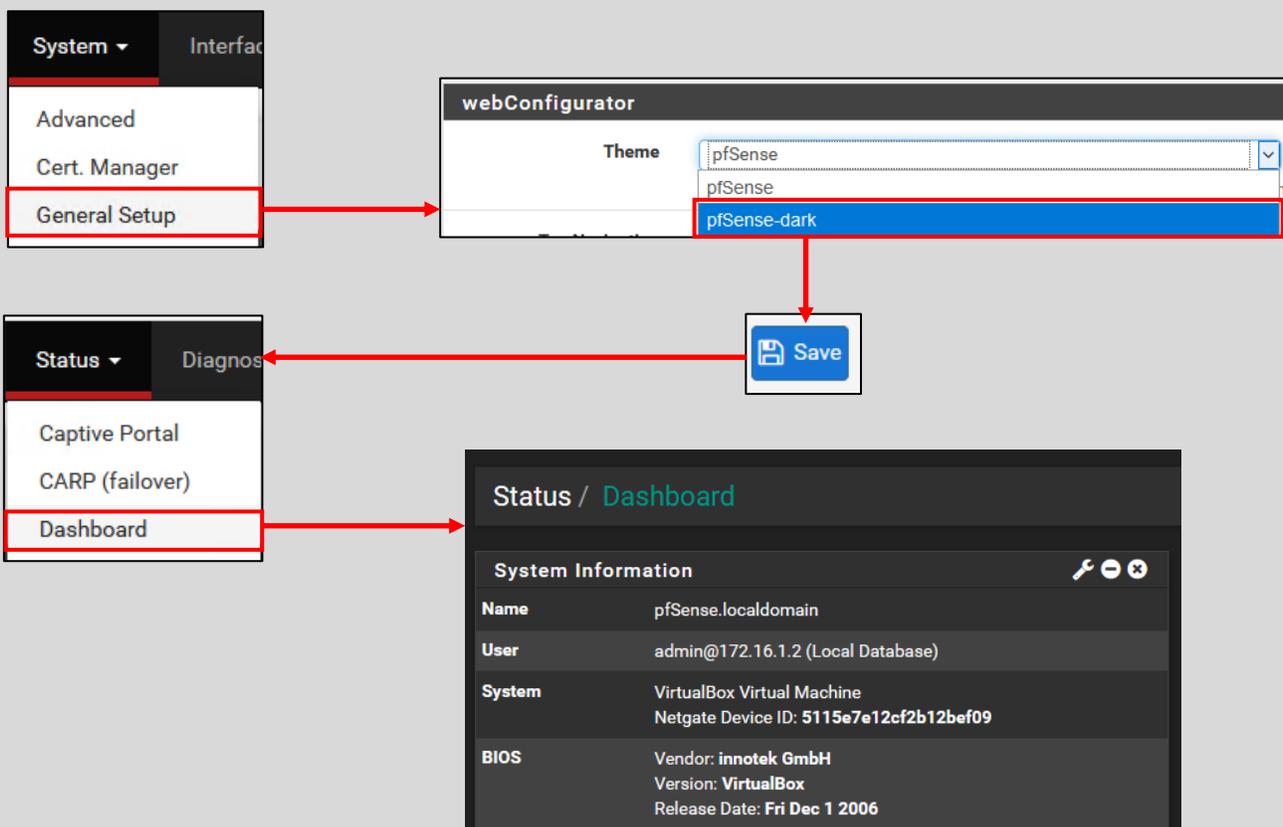
For students who ran into problems with their connectivity checks, or getting an IP address on the WAN interface, you would be led to believe that the *System Update* page would be a great way to test whether or not connectivity is working. In my observations of disconnecting the WAN interface of the VM from a network segment with internet connectivity, then attempting to check for updates, the behavior isn't what you would expect. The page was slower to load, the *Status* field displayed *Retrieving...* with a gold spinning gear, but after a moment, displays *Up to date*, despite having no way to confirm that with the update servers.

The bottom line is, **the *System Update* page is not a good way of confirming whether or not students have internet connectivity.** If you were required to connect the pfSense VM's adapter mapped to the WAN interface to a NAT network, run the *Set interface(s) IP address* wizard to configure the WAN interface manually (e.g., IP address, netmask, default/upstream gateway), or specify a primary and/or secondary DNS server during the *pfSense Setup* wizard, **re-run connectivity checks to confirm the pfSense VM can reach the internet.** If they are still failing, there is still troubleshooting to do. If the commands confirmed the pfSense VM is connected to the internet and the *System Update* page displays the status *Up to date*, congratulations, the system is fully updated, and you are ready to proceed.

The Goggles Do Nothing

Most IT personnel have to sit and stare at screens all day. While there are some solutions that can be used to dim your screen, or modify the color of your display (e.g., f.lux, Windows Night light, MacOS Night Shift, etc.), you can also make use of application themes that use a darker color palette. The webConfigurator has a theme labeled *pfSense-dark* that can be activated. You don't have to, but I recommended activating it alongside your preferred solution for reducing color temperature/brightness in order to help reduce eyestrain.

To activate the theme, Navigate to *System > General Setup*. On the *General Setup* page, there is a section labeled *webConfigurator*. The first field, and the only one we care about is labeled *Theme*. Click the drop-down, select *pfSense-dark*, then scroll to the bottom of the page and click *Save*. The theme will not activate until you navigate away from the *General Setup* page. Any page will work, so for example, select *Status > Dashboard* from the pfSense navigation menu to load the *Dashboard* page. Sweet relief, my eyes are no longer burning. If you noticed the DNS settings on the *General Setup* page, get yourself a cookie and make a mental note – we'll be back here later to fiddle with that.



14-5: This illustration describes the process for activating the pfSense-dark theme. If you value your eyesight, you'll follow these instructions.

14.3 Enabling Network Services

This section will demonstrate how to enable various core network services on the pfSense VM for our lab environment. It will be broken up into 4 subsections – DNS Forwarding, NTP, Squid HTTP proxy, and DHCP. While DHCP services should already be enabled from the *Set Interface(s) IP address* wizard, this section will enable students to fix any mistakes they may have made, and also guide them through creating static DHCP mappings, an important task for configuring the remaining lab virtual machines.

14.3.1 DNS Forwarding

DNS as a network and security subject is a black hole so dense with information and arcane nuance, that it has its own gravitational pull. It is both powerful and unwieldy. A double-edged plasma lance as likely to cut the wielder as it is to serve them. When they say *it's always DNS*, they mean it. In this section, students will learn how to enable DNS forwarding on pfSense, and why this is a service we want for our lab environment

During the pfSense Setup wizard, users are invited to enter a primary and secondary DNS server in order for the pfSense VM to resolve hostnames. The wizard states that those DNS servers will not be used by other DNS clients (e.g., the rest of our lab environment) unless DNS query forwarding is enabled.

By default, DNS queries for other clients are automatically forwarded to the root DNS servers. The best way to describe the root DNS servers is that they are considered the most authoritative source on domain names on the internet. If queried directly, it's likely that you'll always get the truth, but at a cost of repeated recursive queries, and latency required to go from querying root DNS servers, to querying the authoritative DNS server for the domain you are trying to resolve – among many other aspects to consider.

DNS forwarding allows clients using the pfSense VM for DNS services to benefit from multiple layers of caching, and the benefits the specified DNS servers provide. Let's use the DNS server 9.9.9.9 for example. We'll call it the "Quad Damage" DNS server. We enable DNS forwarding on pfSense, and tell pfSense to relay DNS queries to Quad Damage.

When a client makes a DNS request, our pfSense VM can check its local cache to see if it resolved the hostname recently. If so, the IP address for that hostname can be retrieved from the pfSense VM's memory and relayed back to the client. This is a cache hit, and the answer for the query is returned very quickly. Now let's say the hostname our client is trying to resolve is not in the cache. pfSense forwards it to Quad Damage. The Quad Damage DNS server can check its cache to see if it resolved that hostname recently and if it has, relay it back quickly. Now, let's say the hostname the client is trying to resolve is malicious. Quad Damage keeps track of known malware domains and will refuse to resolve them, blocking our infected client from reaching a command and control server. We just benefitted from Quad Damage's cache and/or its malware filtering.

While querying the root DNS servers directly is guaranteed to work because DNS across the internet relies on them, DNS forwarding is a way for us to take advantage of benefits and caching other DNS servers provide. Not only that, we can configure multiple DNS servers to forward our queries to in case Quad Damage is unavailable.

When ready, log in to the pfSense webConfigurator, and navigate to *Services > DNS Resolver*. Locate the field labeled *Network Interfaces*. This setting defines what network interfaces are going to listen for, and respond to DNS requests. **Highlight Localhost, LAN, and OPT1 only.** This makes it to where only the pfSense VM itself, or systems on the *LAN* or *OPT1* networks can use pfSense as a DNS resolver. Different operating systems have shortcut keys to allow users to select multiple items from a menu. For MacOS, this can be done by holding the meta key before left clicking. For Linux and Windows users, usually its the ctrl or shift key.

In the next field, labeled *Outgoing Network Interfaces*, **Highlight the WAN interface only.** This option defines which network interface is used for making outbound DNS requests. The WAN interface should be the only interface connected to a network with any sort of connectivity to an upstream DNS server.

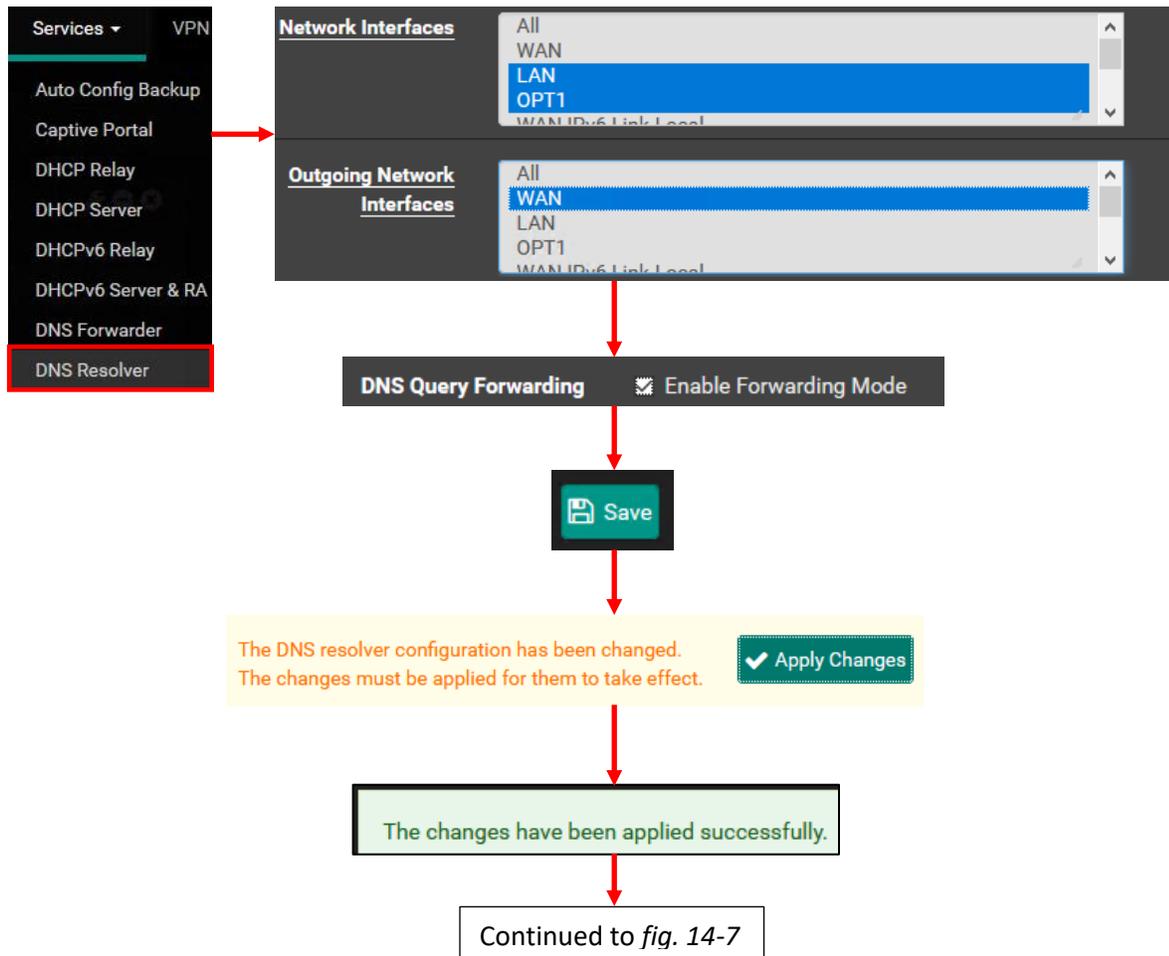
That brings us to our third and final setting for this page. Find the *DNS Query Forwarding* field, and check the checkbox labeled *Enable Forwarding Mode*. On the bottom of the page, click *Save*. Then, when the page reloads, click the *Apply Changes* button. The page will update with the text, *The changes have been applied successfully once completed*.

Next, select *System > General Setup* from the navigation menu. On the *General Setup* page, under the *DNS Server Settings* window, are a handful of different fields, inputs boxes, and checkboxes to control how the DNS resolver operates. Let's start with the field labeled *DNS Servers*, and the field below it, *Add DNS Server*. The *DNS Servers* field has two input boxes, labeled *Address* and *Hostname*, and a drop-down menu labeled *Gateway*. If there are specific DNS servers students want or need to use (you know your local network, and ISP better than I do), put the IP address of the DNS server in the *Address* field. The *Hostname* input box is for setting up DNS over TLS (DoT). For the purposes of our lab environment, DoT is not enabled, so students do not need to worry about this field. The *Gateway* drop-down allows users to use a specific gateway when querying the DNS server. This setting should not need to be modified, and can be left with its default setting of *none*. If users click the *Add DNS Server* button, additional lines appear in the *DNS Servers* field in order to configure more DNS servers. Students may add as many DNS servers to forward queries to as they would like.

One of these checkboxes should look familiar from the pfSense Setup wizard – *Allow DNS server list to be overridden by DHCP/PPP on WAN*, in the *DNS Server Override* field. As previously mentioned, some enterprise networks block systems on the local network from making DNS queries outbound. The idea is that in most of those networks, Microsoft Active Directory is used heavily, and heavily relies on Microsoft DNS to find hosts and services. Clients on the network should be using the DNS services provided by their local DNS server and/or domain controllers

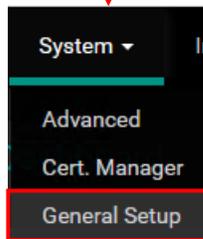
to active directory. Those active directory servers should be configured with Primary/Secondary DNS servers for the site/network that are responsible for forwarding queries for internet hostnames.

In other cases, this checkbox may need to be checked due to how some internet service providers (ISPs) operate, requiring the use of their DNS servers for internet hostname resolution. Students will know their local networks and ISP requirements best, and will likely know whether or not they need this option checked. If do not have to use specific DNS servers supplied by DHCP (or you have to manually set them yourself) uncheck this option, and following the instructions in the previous paragraph for adding DNS servers. Once you have finished, click *Save*.



14-6: To enable DNS services, and DNS forwarding, Navigate to *Services > DNS Resolver*. ***In the Network interfaces field, highlight LAN, OPT1, and Localhost. For the Outgoing Network Interfaces field, ensure that only the WAN interface is selected.*** Scroll down, and check the *DNS Query Forwarding checkbox*. Finally, Click the *Save* button, then the *Apply Changes* button.

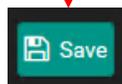
Continued from *fig. 14-6*

The 'DNS Server Settings' page in pfSense. It features a table with three rows for DNS servers. The first row has '8.8.8.8' in the 'Address' field and 'no' in the 'DNS Hostname' field. The second row has '4.2.2.2' in the 'Address' field and 'no' in the 'DNS Hostname' field. The third row has '9.9.9.9' in the 'Address' field and 'no' in the 'DNS Hostname' field. Below the table is an 'Add DNS Server' section with a '+ Add DNS Server' button. At the bottom is a 'DNS Server Override' section with a checkbox labeled 'Allow DNS server list to be overridden by DHCP/PPP on WAN'.

DNS Servers	Address	DNS Hostname	no
	8.8.8.8		no
	4.2.2.2		no
	9.9.9.9		no

Add DNS Server + Add DNS Server

DNS Server Override Allow DNS server list to be overridden by DHCP/PPP on WAN
If this option is set, pfSense will use DNS servers assigned by a DHCP Resolver). However, they will not be assigned to DHCP clients.



The changes have been applied successfully.

14-7. Next, navigate to *System > General Setup* page on pfSense in order to modify *DNS Server Settings*. Input the IP address of DNS servers to use in the *Address* input box. Users can create more DNS server entries by clicking the *Add DNS Server* button. Unless students have a specific need to use DNS servers assigned via DHCP, uncheck the *DNS Server Override* checkbox in order for client systems (read: our lab VMs) to be able to make use of the DNS servers configured on this page.

What DNS Servers Should I use?

There are a wide variety of options available for public DNS servers on the internet. Many of them are ubiquitous (for example, Google's 8.8.8.8 and 8.8.4.4 are well-known DNS servers), while many more offer access control, and filtering of known malware domains (e.g., Cisco's OpenDNS, among many others). Still yet, others say use the DNS servers your ISP provides because they are usually the geographically closest to you (less latency) and cache common responses for other clients of your ISP – this usually means they're pretty fast, but sometimes aren't very reliable when your ISP plans outage windows and doesn't notify customers.

Keep in mind that performance, features, and availability are not the only thing to consider when choosing a public DNS server to forward your queries to. Most public DNS servers have privacy policies stating that domains queried are kept for some period of time by them, but don't specify if the logs are then destroyed, or handed to third parties. Others straight-up tell you that they hold on to the queries and sell that information. In fact, if you live in the United States, the FCC says your ISP is legally allowed to sell that information.

The queries are usually collected into a product called a passiveDNS database. The idea is that researchers can query a database for a domain and see when the domain was first created vs. when it first started seeing queries. Some even allow you to see related domains, domains on adjacent IP addresses, subdomains, parent domains, volume of requests over time, etc. Most malware domains are short-lived, and created shortly before seeing a massive spike in queries before they are either filtered, taken down, or the malware authors change to a new domain. In other cases, passiveDNS may be used by companies to determine success of a marketing campaign, etc.

The question of what public DNS servers to use comes down to balancing features and availability against how much you value your privacy. How much information are you comfortable with that public DNS server having on you, your network, and your clients? They will promise you that they delete data after a certain period of time, or only log certain portions of the DNS data, but there is no effective way for you to verify that. Remember the popular mantra, if you are not the customer, then you are the product.

Consider it a homework assignment to look up the privacy policies and/or performance of various public DNS server providers:

- Google DNS
- Open NIC
- Quad 9
- Cloudflare
- DNS Watch
- Your ISP's DNS servers

What about DNS over TLS (DoT), DNS over HTTPS (DoH), DNSSEC, and DNSCrypt?

Because this section is already much longer than I had intended it to be, I'm going to *briefly* talk about some relatively recent technologies attempting to make DNS more secure in some way. **Setting these up are outside of the scope of what's covered in this book**, but I felt it important to at least touch upon them.

DNSSEC addresses some integrity concerns with DNS. In a nutshell it requires someone to sign their DNS data (the same way we are required to sign SSL certificates and software drivers through a recognized authority, for example) to attest that the information returned is accurate and has not been tampered with. It sounds great, but it's not yet widely adopted, and DNS experts are a little leery about tying Integrity and Authenticity of DNS records to certificate verification infrastructure. That is a huge can of worms that I'm not going to delve any deeper into, otherwise this section would be much longer than it needs to be. DNSSEC does absolutely nothing to keep DNS queries confidential in transit – it does not encrypt DNS traffic at all. By default, pfSense utilizes DNSSEC if a domain provides it, and I recommend leaving it enabled.

DNSCrypt provides encryption to protect DNS queries from interception/tampering in transit, but is not widely implemented. Many consider DNSCrypt to be deprecated by DNS over TLS (DoT) and/or DNS over HTTPS (DoH).

DNS over TLS is DNS wrapped in SSL over its own custom port. This provides the same in-transit security that DNSCrypt provides, but it's an actual standard, a bit more widely adopted, and more widely supported. A lot of popular free DNS providers provide a DoT server. The fact that it uses its own service port (853/TCP) is considered by many to be a hindrance, or a blessing depending on who you ask. Why? Requiring its own unique service port makes it easy to block. For security engineers and analysts, mitigating possible threats that use DoT is as simple as dropping packets in or outbound on port 853/TCP. More privacy conscious individuals see this as a disadvantage because of how easily the service can be suppressed.

DNS over HTTPS is DoT's big brother. It has a design specification, and a lot of support from really big tech companies. I absolutely despise it from the perspective of a network defender. So, if you're expecting an unbiased view on what DoH is, this is not the place.

Users make encrypted web requests to a DoH server. These requests are in the form of serialized JSON either over HTTPS or HTTP/2. The server then gets the response, and either can satisfy the request from cache, or does recursion to other DNS servers. This is already several layers of encapsulation and/or parsing if an analyst wanted to inspect DoH traffic for either troubleshooting, or malicious use.

The requests and responses look like HTTPS traffic, and it uses port 443 just like encrypted web traffic. Hypothetically, it could be blocked by disabling inbound/outbound access over port 443/TCP, but you would be blocking over 50% of the web while you were at it. According to security company ESET and their *we live security* blog, as of 2018, over 50% of the top 1 million websites are served over HTTPS, and that number continues to grow every year.

The fact that DoH piggybacks off of such a commonly used protocol makes it very difficult to detect and/or filter. This is considered a win for privacy advocates on the internet, but means a massive loss in visibility for network defenders and engineers as a whole. No more passive DNS, no more using Wireshark to troubleshoot DNS issues. No more malware domain blocklists because network intrusion software, proxies and firewalls that cannot do SSL decryption (which is a much bigger can of worms, and very difficult to do correctly) can't block malicious DoH requests. Defenders can create network policies to disable DNS over HTTPS on operating systems and/or web browsers. They can also choose to blocklist known DNS over HTTPS servers, but all an adversary would need to do is configure their own DoH server(s) and code that into their malware.

Hypothetically, DoH could be blocked by blocking both the IP addresses and domains associated with known DoH services, but that's not a practical solution (at least on its own), as more people and organizations stand up their own DoH servers.

In addition to being a huge blind spot for defenders, there are also privacy trade-offs. DoH uses HTTPS or HTTP/2. There is a lot of metadata that gets sent with those DoH requests in the form of HTTP headers. There is nothing stopping an organization that provides a DOH server from logging all of this additional information, and the users would never know (unless it was disclosed by the organization running the server). This was a subject of great contention some years ago when former NSA contractor Edward Snowden leaked classified information, unveiling the existence of the PRISM program. PRISM unveiled that the Intelligence Community partnered with several big tech companies to gain access to sensitive information for national security purposes. This sparked a lot of outrage, and a wave of mistrust in American tech companies at the time. It seems that the internet is quick to forget past transgressions, because DNS over HTTPS is very opaque and enables that kind of abuse with no way for the users to know that it is happening all over again.

These are just the issues I can think of off the top of my head. There are many more that I won't bore you with. The bottom line is, **I don't recommend DNS over HTTPS at all.** But at the end of the day, this is your lab environment and your choice if you'd like to pursue it further.

14.3.2 NTP

NTP, or Network Time Protocol, is a means for computers to keep accurate time. In my personal experience, it is best described as a malevolent little gremlin by which things assumed to be well-oiled will burst into flames at a moment's notice. A lot of programs and services rely on accurate timekeeping. For example, SSL certificates are issued with a start date and end date that determines how long they are considered valid. If your system's internal clock has drifted too far forward, or too far back, some SSL certificates may no longer be considered valid. This problem can manifest itself in a number of ways – all because the system's time is off.

Digital forensics and incident response (DFIR) is another reason why accurate timekeeping is important. Most logs include a timestamp of when events occurred. If an incident or attack occurs, accurate logs and timestamps become extremely important for building an event timeline, and/or the scope of an incident. In this section, students will configure the pfSense NTP server.

To begin, select *Services > NTP* from the pfSense navigation menu to bring up the *Settings* page for NTP. We are interested in the *Interface*, *Time Servers*, and *Add* fields/button in the *NTP Server Configuration* window. The *Interface* selection menu operates very similarly to the ones encountered in section 14.3.1. This field determines which interfaces will respond to NTP requests. NTP should only be available virtual machines in the lab environment, so **select and highlight the LAN and OPT1 interfaces only.**

By default, the *Time Servers* field has an input box that should be pre-filled with the hostname `2.pfsense.pool.ntp.org`, with three checkboxes next to it. The only checkbox that should be checked is the one labeled *Is a Pool*. This checkbox is the only one we're really concerned with, because it indicates that the NTP server entered is actually a collection of NTP servers.

The default NTP server pfSense provides should be sufficient for the needs of most students, but if desired, the default entry can be modified, or additional NTP servers can be configured through the *Add* button on the page. This may be required for users joined to networks where external NTP traffic is prohibited, and internal NTP servers are provided, instead. In other cases, it may be a good plan to specify other NTP servers based on the region of the world readers live in. Enter the NTP pool project.

Open up a browser and navigate to `www.ntppool.org`. This website is dedicated to tracking NTP pools all over the world. Click the link on the left side of the page labeled, *How do I use pool.ntp.org?* This page has a section with the text:

```
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
server 3.pool.ntp.org
```

If students wanted to use these NTP servers on their pfSense VM (I highly recommend it), start by removing the text from the first *Time Servers* input box (the pfSense NTP pool), click the *Add* button three additional times then, enter the following:

Input box 1: 0.pool.ntp.org

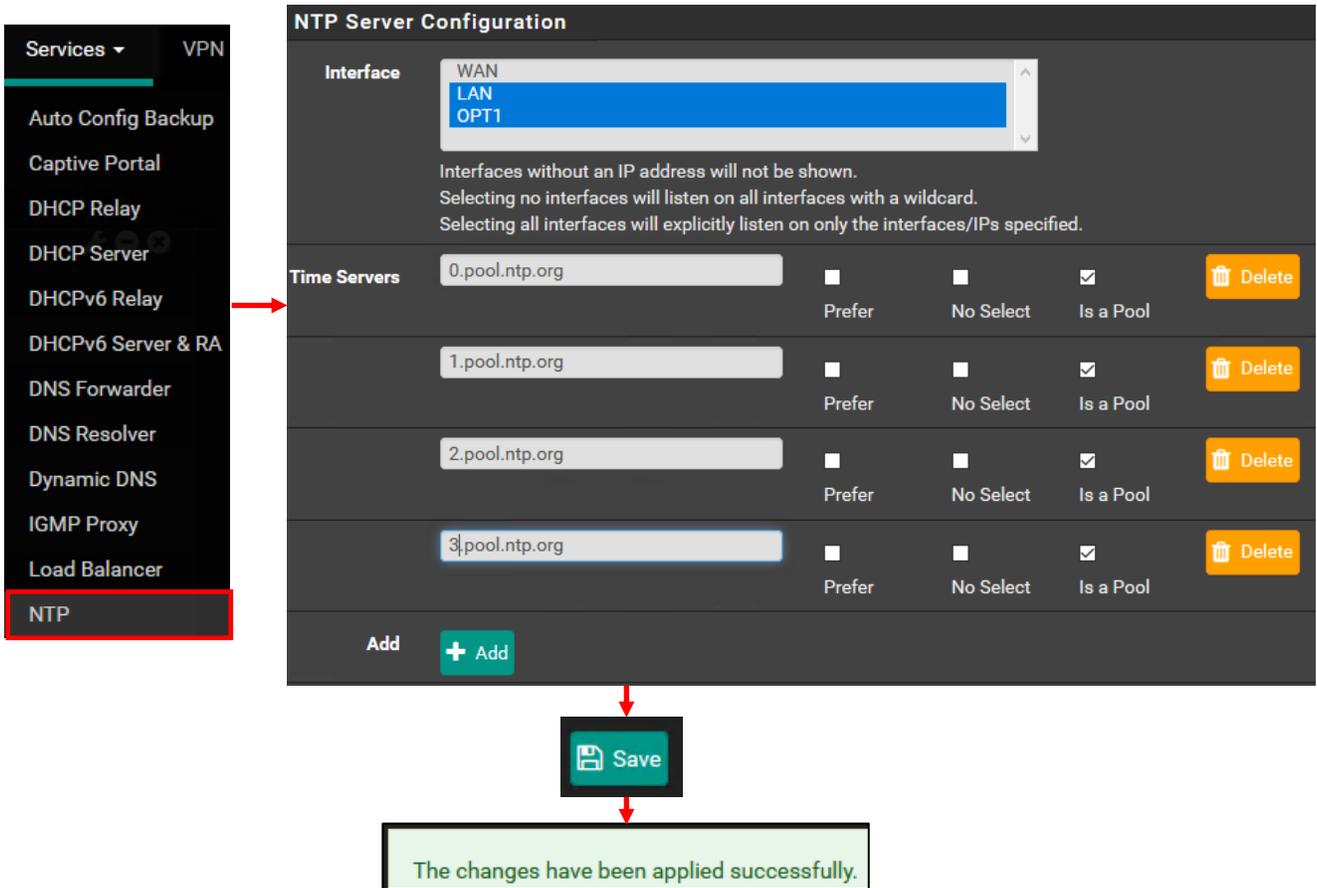
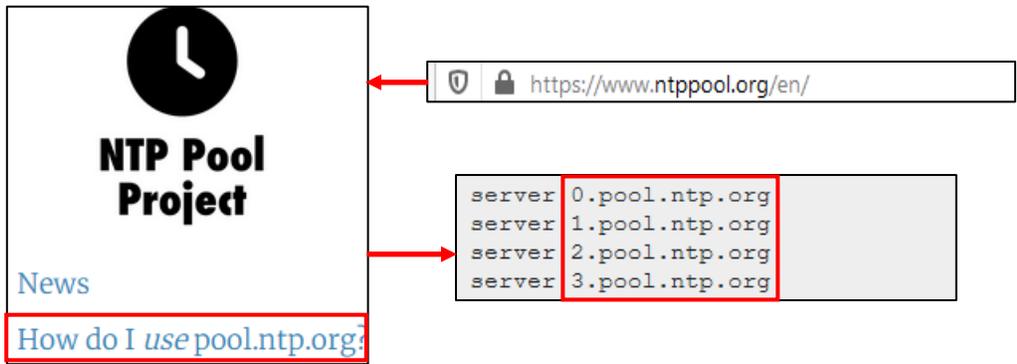
Input box 2: 1.pool.ntp.org

Input box 3: 2.pool.ntp.org

Input box 4: 3.pool.ntp.org

Ensure the checkbox labeled *Is a Pool* is selected for each of these entries. If students are joined to networks where they are expected to use private NTP servers, the process is more or less the same. Make sure the default NTP server entry is removed, enter the preferred NTP server IP address or hostname, then add additional NTP servers via the *Add* button. The only difference is that, more than likely, the *Is a Pool* checkbox should be unchecked. Ask network admins for NTP server addresses, and whether or not they are pooled to confirm. After selecting the correct interfaces and entering preferred NTP servers, scroll to the bottom of the page, then Click *Save*.

Note: Setting up NTP on the pfSense is just half of the overall process. Lab virtual machines need to have `ntpd` installed, and be configured to sync to the NTP service on the pfSense VM. Check out Chapter 20, [section 20.3](#) (*pp.* 1080-1084) for the details.



14-8: To find more NTP servers, students should open their preferred web browser, and navigate to www.ntppool.org. Click the link on the left side of the page labeled, *How do I use pool.ntp.org?* On the next page, in a text box in the center of the screen will be a list of NTP servers students can configure on the pfSense VM. Back on the pfSense webConfigurator, access the NTP server settings from *Services > NTP*. Highlight *LAN* and *OPT1* in the *Interface* field. Users can overwrite the default *Time Servers* entry, and/or click the *Add* button to configure additional NTP servers. In this diagram, the global NTP server pool hostnames have been configured. If students are using NTP servers from the NTP pool project, be sure to check the *Is a Pool* checkbox next to each entry. Otherwise, for students required to use private/enterprise NTP servers, the *Is a Pool* checkbox (more often than not – check with your system/network admins) should be unchecked. Click the *Save* button to save any changes made to the NTP service.

14.3.3 Squid HTTP Proxy

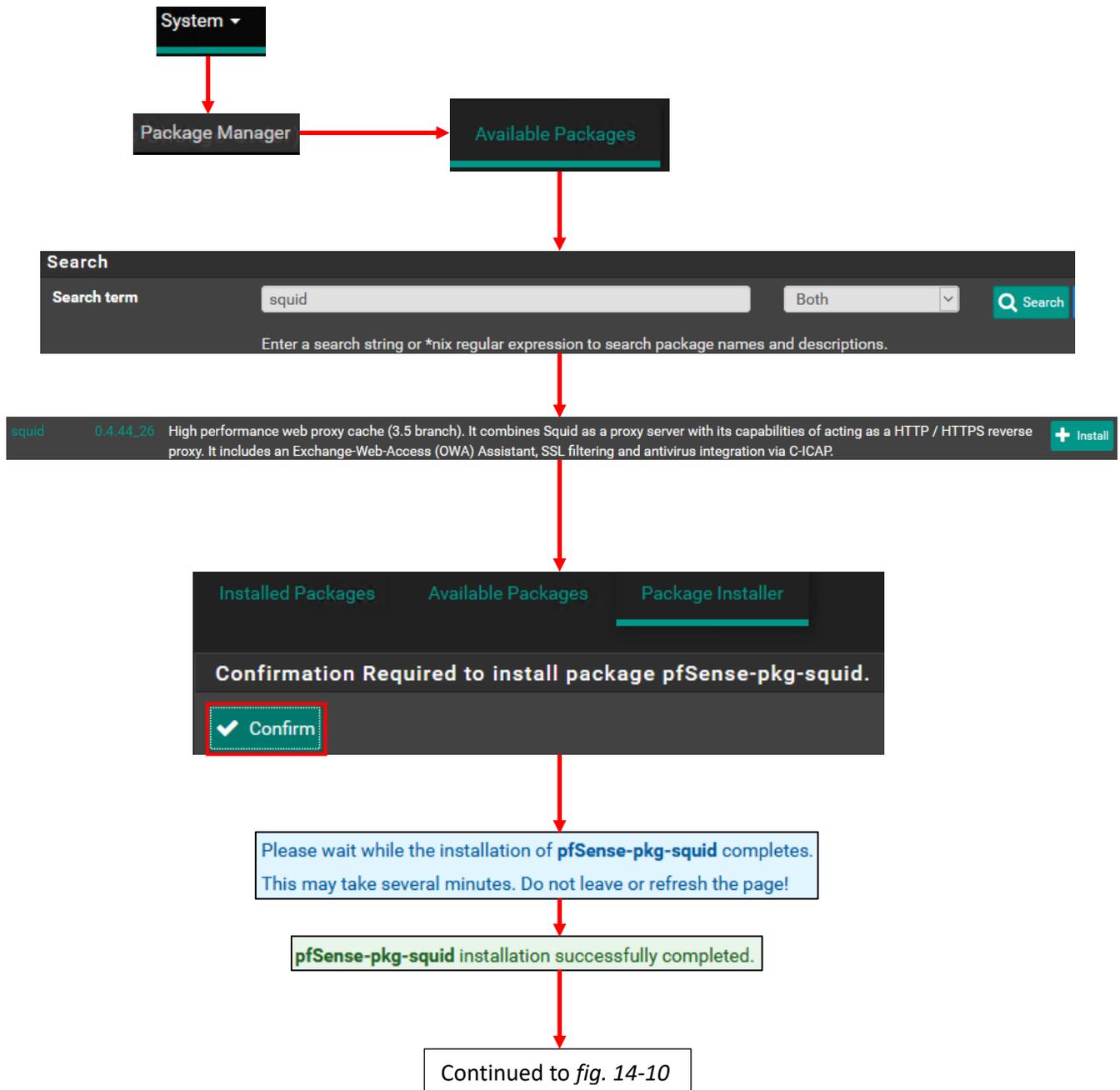
HTTP Proxies serve a variety of purposes. They can be used to allow or deny access to certain websites or URLs, or speed up internet access by saving frequently requested files (caching), among other purposes. In this section, students will utilize the pfSense package manager to install the SQUID proxy server, then configure the proxy server for use in our lab network.

To begin, select *System > Package Manager* from the pfSense navigation menu. Click on the tab labeled *Available Packages*, and in the *Search term* input box, enter *squid*. Hit the enter key or click the *Search* button, and a list of software package names appears in the *Packages* window below. Select the package named *squid*, with the description *High performance web proxy cache (## branch)...*, and click the *Install* button. The Package Installer page opens. Click the *Confirm* button to begin the installation process for the software package. After a moment or two, a confirmation message pops up to confirm SQUID installed successfully. Students should be able to select *Services > Squid Proxy Server* from the pfSense navigation menu. This brings students to the *General* settings page for the squid proxy server. Click on the tab labeled *Local Cache*. pfSense requires the settings on this page be confirmed before allowing users to enable the squid proxy service on the *General* settings tab. The defaults pfSense selects for the local cache are pretty reasonable, so all that needs to be done is scroll to the bottom of the page, click *Save*, then click on the *General* tab.

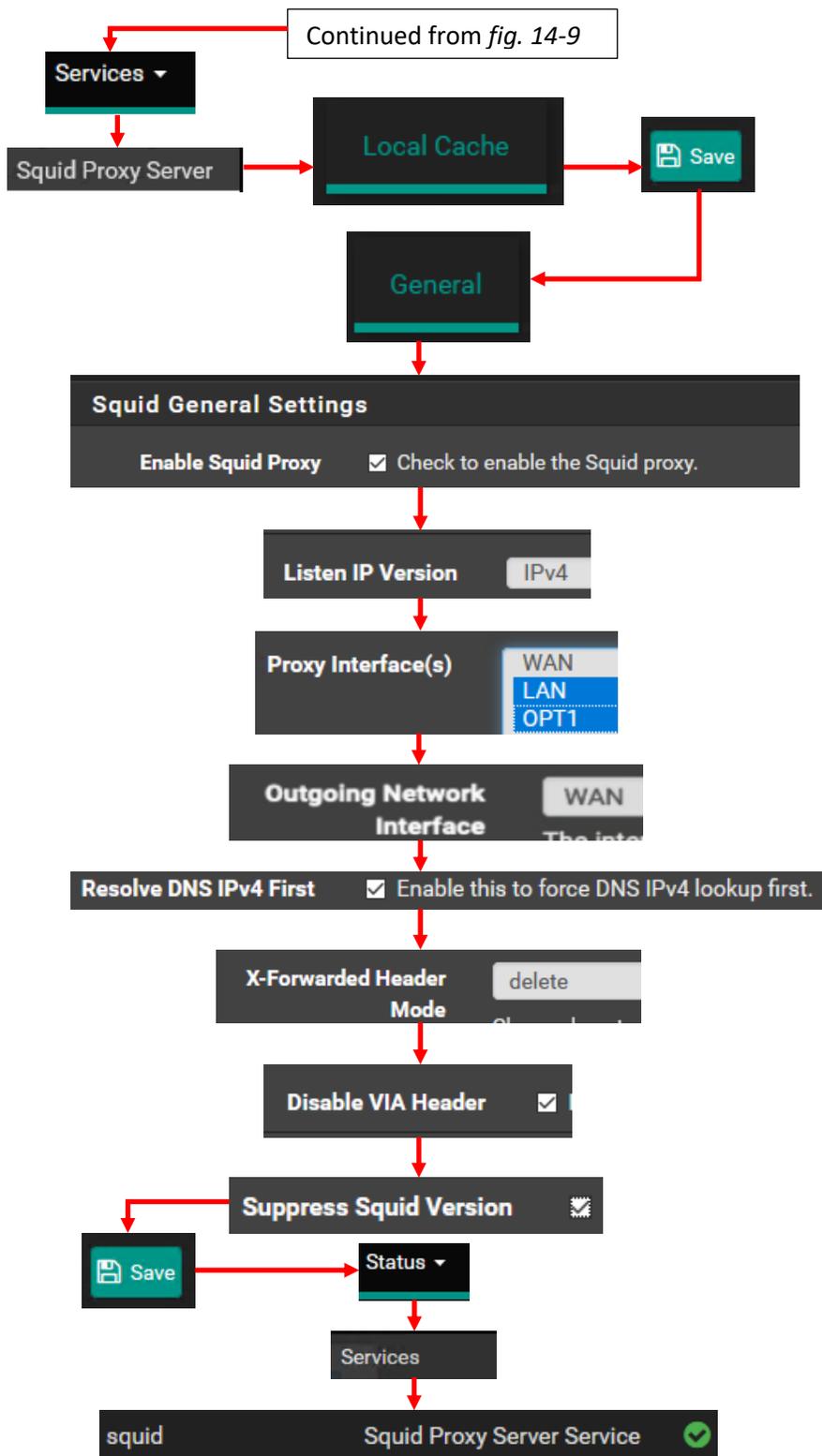
Under the *Squid General Settings* window, click the checkbox in the field *Enable Squid Proxy*. Next, in the *Listen IP Version* field, ensure IPv4 is selected. Next, in the *Proxy Interface(s)* field, highlight the *LAN* and *OPT1* interfaces. Then, in the *Outgoing Network Interface* field, ensure that the *WAN* interface is selected. Afterwards, click the checkbox in the field *Resolve DNS IPv4 First*.

Next, scroll down to the Window labeled *Headers Handling, Language and Other Customizations*. In the field labeled *X-Forwarded Header Mode*, select the *delete* option from the drop-down menu. Next, click the checkbox in the *Disable VIA Header* field. Finally, click the checkbox in the field labeled *Suppress Squid Version*. These are all fields that leak metadata, or information about the virtual lab that external hosts do not need to know about. After making all of these changes, Click the *Save* button at the bottom of the page. The page will not reflect whether or not the squid proxy service is running. If students are interested in seeing the status of various services on the pfSense VM (including squid), select *Status > Services* on the navigation menu.

Note: Our lab environment is primarily IPv4. If students have any plans to experiment with IPv6 support, be sure to change the *Listen IP Version* field under *Squid General Settings* to either *IPv6*, or *IPv4+IPv6*.



14-9: Navigate to *System > Package Manager* and click on the *Available Packages* tab. Search for *squid*, and install the squid web proxy package.



14-10: After installing squid, navigate to *Services > Squid Proxy Server*. Click on the *Local Cache* tab, then click *Save* to accept the default cache settings. Navigate back to the *General* tab, check *Enable Squid Proxy*, ensure *Listen IP Version* is set to *IPv4*, set *LAN* and *OPT1* as the *Proxy Interface(s)*, set the *WAN* interface as the *Outgoing Network Interface*, check *Resolve DNS IPv4 First*, set *X-Forwarded Header Mode* to *delete*, check both the *Disable VIA Header*, and *Suppress Squid Version* checkboxes, then click *Save*. Next, select *Status > Services*, and verify the *Squid Proxy Server Service* has a green checkmark.

14.3.4 DHCP

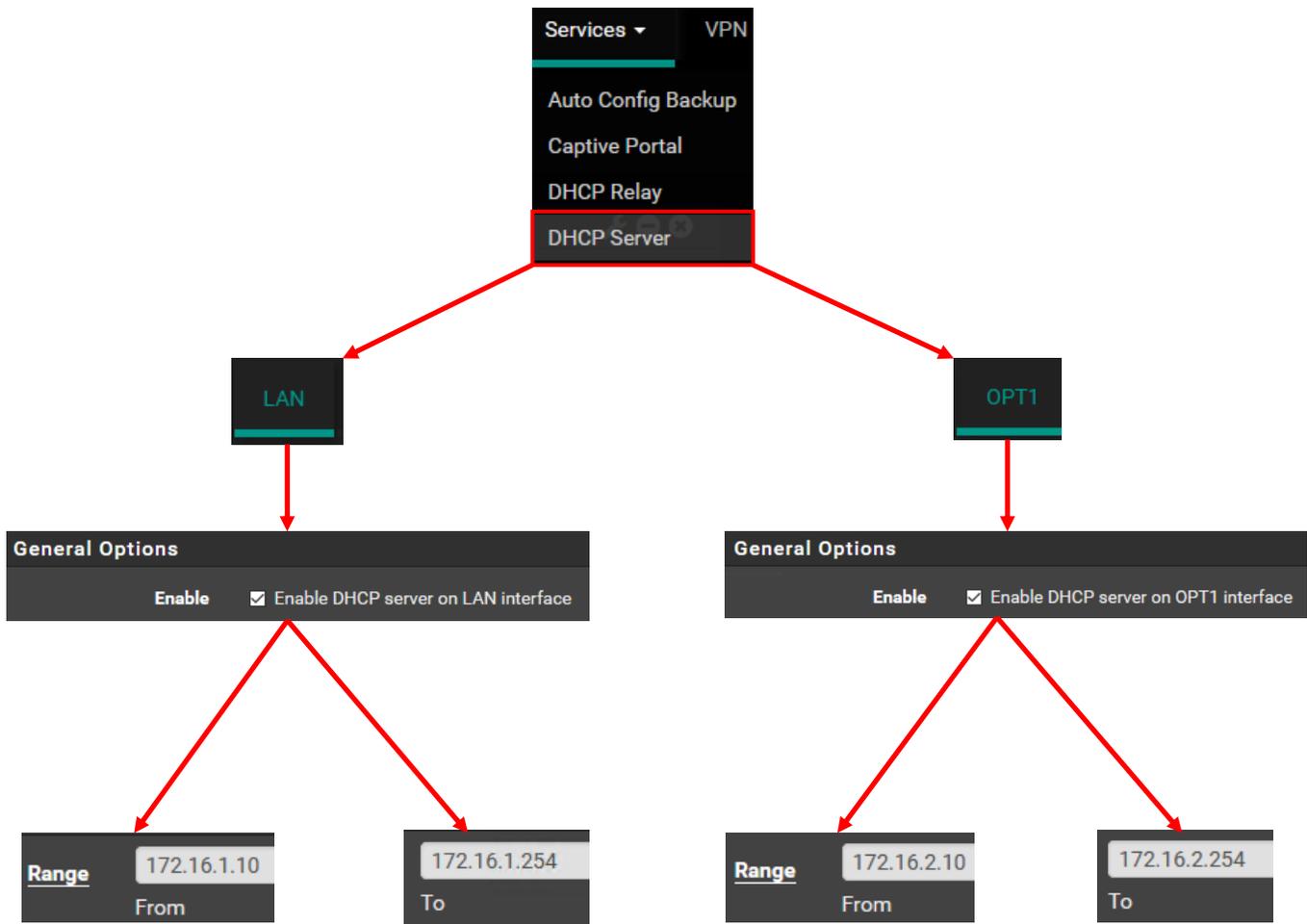
Before being directed to come to this chapter from your selected hypervisor setup guide, students were guided on configuring IP addresses for the *WAN*, *LAN* and *OPT1* interfaces on the pfSense command-line menu, through the *Set interface(s) IP address* wizard. Part of the wizard asks if users are interested in setting up DHCP on the interface, and if so, the start and end IP address for the pool of IP addresses DHCP is allowed to assign. This pool of addresses is referred to as a DHCP scope. For example, the scope for the LAN interface (management network) is 172.16.1.10 – 172.16.1.254, while the OPT1 interface (IPS 1 and IPS 2 networks) scope is 172.16.2.10 – 172.16.2.254.

In this section, students will learn how to navigate the DHCP Server settings page in pfSense, in order to customize DHCP settings and scope as needed. Additionally, they will learn how to create static DHCP mappings to make sure our baseline lab virtual machines always obtain the same IP address from the DHCP server. To get started, select *Services > DHCP Server* from the pfSense navigation menu.

The DHCP server settings page appears, and defaults to displaying the configuration for the *LAN* interface and network. There are tabs that allow users to switch between the *LAN* and *OPT1* interface settings near the top of the page. On both the *LAN* and *OPT1* pages under the *General Options* section, the checkbox in the Enable field, titled *Enable DHCP server on [LAN|OPT1] interface* should already be checked. If it's not click the checkbox and enable the DHCP service on both the *LAN* and *OPT1* networks.

Next, take notice of the *Range* field. This field contains two checkboxes labeled *From* and *To*. If students followed the instructions from the hypervisor setup guide, the *LAN* interface should have 172.16.1.10 in the *From* input box, and 172.16.1.254 in the *To* input box. Likewise, the *OPT1* interface should have 172.16.2.10 in the *From* input box, and 172.16.2.254 in the *To* input box. If this is not the case (and you are NOT using an alternate network range to avoid IP address conflicts), enter those values now, and click *Save*.

Note: If you are using an alternative network range, I recommended in order to avoid or work-around network conflicts, I also suggested start and end addresses for those DHCP scopes. Hop back to your hypervisor setup guide, and look for sidebar discussion: *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range?* Compare the *Subnet*, *Subnet mask*, *Available Range*, and *Range* fields on the DHCP settings page for your LAN/OPT1 interfaces to the suggestions I provided, and correct them where necessary. If you decided to implement your own custom IP address, netmask and DHCP scope, my general guidance is to not assign the first ten IP addresses of the subnet to your scope, so they can be used for static DHCP allocations.



14-11: This illustration demonstrates how to access the DHCP server settings page on the pfSense webConfigurator. Once there, we are interested in confirming that DHCP is enabled on both *LAN* and *OPT1*, and verifying the *from* and *to* IP addresses in the *Range* field are correct on both networks, to ensure the DHCP server has proper scopes configured.

14.3.4.1 How to Create a Static DHCP Mapping

Way back in Chapter 6, the lab network diagram specified what IP addresses each device will have on each network segment. You could opt to statically configure each of our virtual machines to ensure they keep the same IP address consistently, but why do that, when the DHCP server can do it for us? The pfSense DHCP Server can reserve an IP address in its network range, so long as its outside of the configured DHCP scope (but within the same subnet), and make sure that a defined MAC address always gets the same IP address. This is called a static DHCP allocation, or mapping.

For example, the OPT1 network (IPS1 and IPS2) is assigned the network 172.16.2.0/24. This network has 256 possible IP addresses (determined by the subnet mask of 24). We take away two of these address – 172.16.2.0 (the network address), and 172.16.2.255 (the network broadcast address). So, the range of assignable addresses is 172.16.2.1 to 172.16.2.254 (If you were paying attention on the DHCP Server settings page, on the OPT1 tab, this matches up with the Available Range field). Students were instructed to set the DHCP scope for the OPT1 network from 172.16.2.10 to 172.16.2.254. Why are 172.16.2.1 through 172.16.2.9 not included in the scope? Well, 172.16.2.1 belongs to the OPT1 interface itself, and we can't let the DHCP server reassign that. That leaves us 8 addresses outside of the scope, and 8 possible static DHCP mappings. Students will be taking another two of those available mappings for the Kali (172.16.2.2) and Metasploitable 2 (172.16.2.3) VMs, leaving students with 6 more possible static mappings – in case students wanted to add more virtual machines later, and needed them to get assigned a consistent IP address.

What can students do if 6 is not enough and they would like to add more VMs with static DHCP allocations? Reconfigure the start IP address of the DHCP scope. For example, if readers wanted 9 more possible static DHCP allocations, the OPT1 DHCP scope *From* address could be reconfigured to 172.16.2.20, instead of 172.16.2.10.

So that explains what a static DHCP allocation is, and the purpose they serve, but how are they configured? **Students need the MAC address of the network interface they wish make an allocation for. This is yet another reason for students to record the MAC address, and network segment for each network interface of each Virtual Machine.** Having that information on-hand is convenient from an asset management perspective, and makes assigning static DHCP allocations easier.

Every hypervisor has a different way of assigning MAC addresses – VirtualBox will assign the MAC address of each enabled virtual network adapter as soon as its created, and/or will allow users to manually edit the MAC address. Client Hyper-V and VMware hypervisors are slightly different in that users can manually specify a MAC address if they want, or wait until first boot for the hypervisor to randomly assign one for them. **If students have come back to this section from their hypervisor guide to do static DHCP allocations, they should already know how to record the MAC addresses of every network interface of each of their virtual machines. Instructions**

have been provided on when and how the MAC address can be recorded for each VM, in each of the hypervisor setup guides. In the example below, We are going to assign the MAC address 20:16:06:DA:66:70 to the IP address 172.16.2.9 on the OPT1 DHCP configuration page.

Navigate to *Services > DHCP Server* and click on the *OPT1* tab. Scroll all the way to the bottom of the page, to the section labeled *DHCP Static Mappings for this Interface*. In the bottom-right corner, click the *Add* button. This opens the page titled, *Edit Static Mapping*, with a main window labeled *Static DHCP Mapping on OPT1*. There are three fields, and their corresponding input boxes we're interested in: the MAC Address, IP Address and Description fields. In the MAC address field, enter the MAC address you want to create a mapping for. The MAC address will need the colon character (:) placed every two characters in this field. For instance, VirtualBox would display our example MAC address as 201606DA6670. This MAC address would need to be entered as 20:16:06:DA:66:70. The IP address field defines what IP address we want to have mapped to the entered MAC address. In this case, we'll enter 172.16.2.9 in the input box. Optionally, students may wish to enter something in the Description field's input box to serve as a reminder what purpose their static DHCP mapping serves. Everything else on this page can be left blank. Scroll to the bottom of the page, click *Save*, then *Apply Changes*. Scroll back down to the bottom of the OPT1 DHCP server settings page to see the new static DHCP mapping. Notice the trash bin icon and pencil icon in the lower right corner of the entry. These icons allow users to delete or edit individual static mappings.

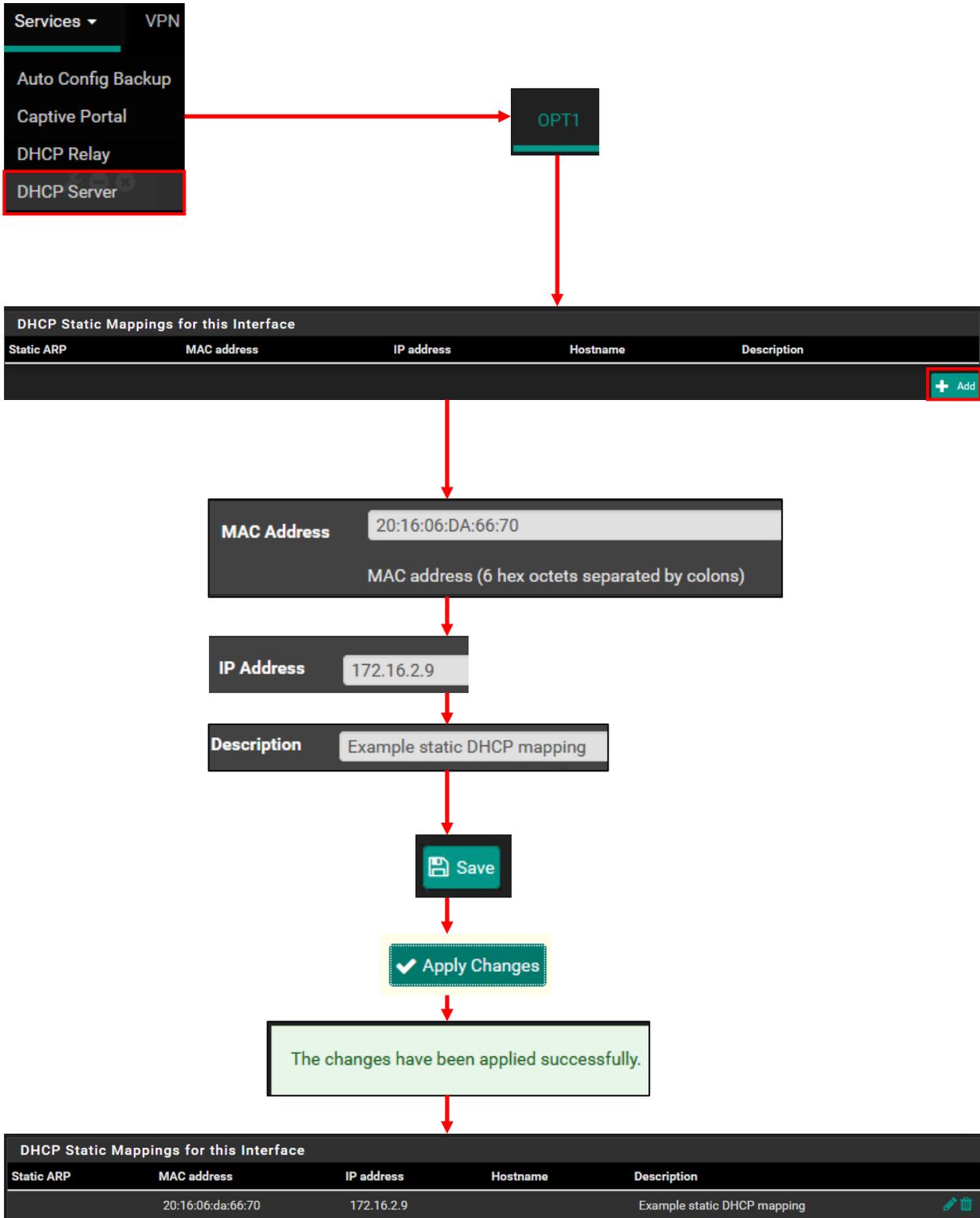
Students will assign a minimum of 4 static DHCP mappings for their lab environment in total:

LAN static DHCP mappings:

172.16.1.3: SIEM
172.16.1.4: IPS

OPT1 static DHCP mappings:

172.16.2.2: Kali
172.16.2.3: Metasploitable 2



14-12: This diagram illustrates how to create a static DHCP mapping on the OPT1 interface DHCP settings. All students need is an IP address (outside of the DHCP scope) and MAC address. A description is highly recommended. This process is identical for the LAN interface DHCP settings.

14.4 Firewall Policy

pfSense is a firewall distribution, and thus far, the firewall functionality has not been discussed, nor configured. In this section, students will learn the basics of network firewalls – specifically, concepts such as what stateful firewall is, firewall rule order, and its effects on network traffic (including the concept of implicit deny any). Afterwards, students will learn about pfSense firewall aliases, and create one to represent RFC1918 networks. Students will then learn how to use the pfSense webConfigurator to create and rearrange firewall rules.

Finally, students will be provided a screen capture of what their firewall policies for the LAN, WAN and OPT1 networks should look like on hosted hypervisors, and a separate set of captures to define what those policies should look like for bare-metal hypervisors. Each of those subsections will include specific directions on deleting the pfSense anti-lockout rule with the express goal of restricting access to the webConfigurator to a smaller number of IP addresses.

14.4.1 Firewall basics – Stateful Firewalls, Rule Order, and Implicit Deny Any

Most modern firewalls are described as being stateful. A stateful firewall is capable of keeping track of connections and determining whether or not packets being transmitted are related to an established connection. The most basic form of this is tracking network sessions by network protocol and port. If a firewall sees a connection between a source/destination IP address pair, and the first packets are allowed for establishing that session, then the other related packets to that established session will also be allowed, until the session is complete, or the connection times out. More advanced stateful firewall technology can allow or deny traffic based on different network application protocol specifications, which users are logged in at specific IP addresses or network(s), and/or the detected operating system of the hosts making network requests. Our firewall rules are going to be fairly basic, based on source/destination IP addresses and/or networks, Protocol (TCP, UDP, ICMP, etc.) and destination port (e.g., 22, 443, etc.).

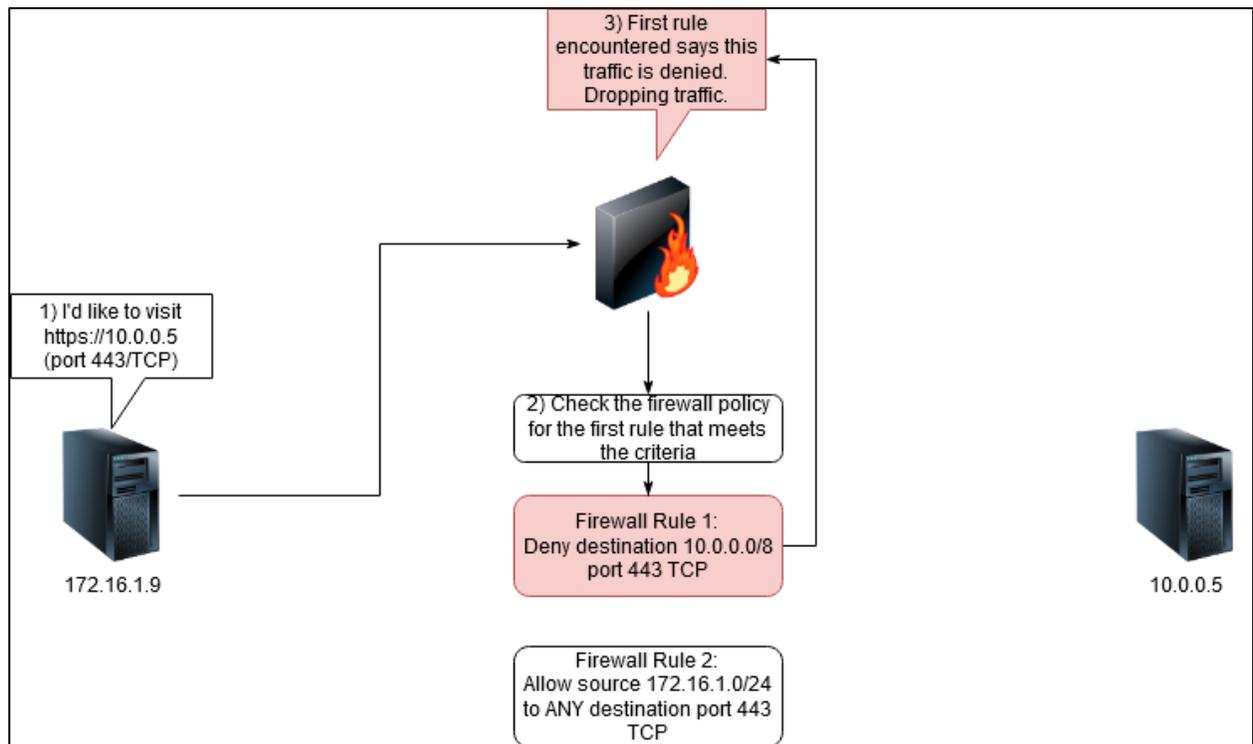
Speaking of firewall rules, it is extremely important to note that **the order in which firewall rules are placed absolutely matters**. Most firewalls will process rules as they apply to traffic from top to bottom. The first rule that matches the nature of the traffic best will be applied first, and the traffic will be allowed or denied based on the action of that firewall rule. Let's go over some examples to help build understanding.

Let's say there are two firewall rules. One rule allows port 443 TCP from network 172.16.1.0/24 to any destination. Then, another firewall rule blocking port 443 TCP to say, network 10.0.0.0/8. Depending on what order these rules are created and/or placed, the effect on network traffic could be wildly different.

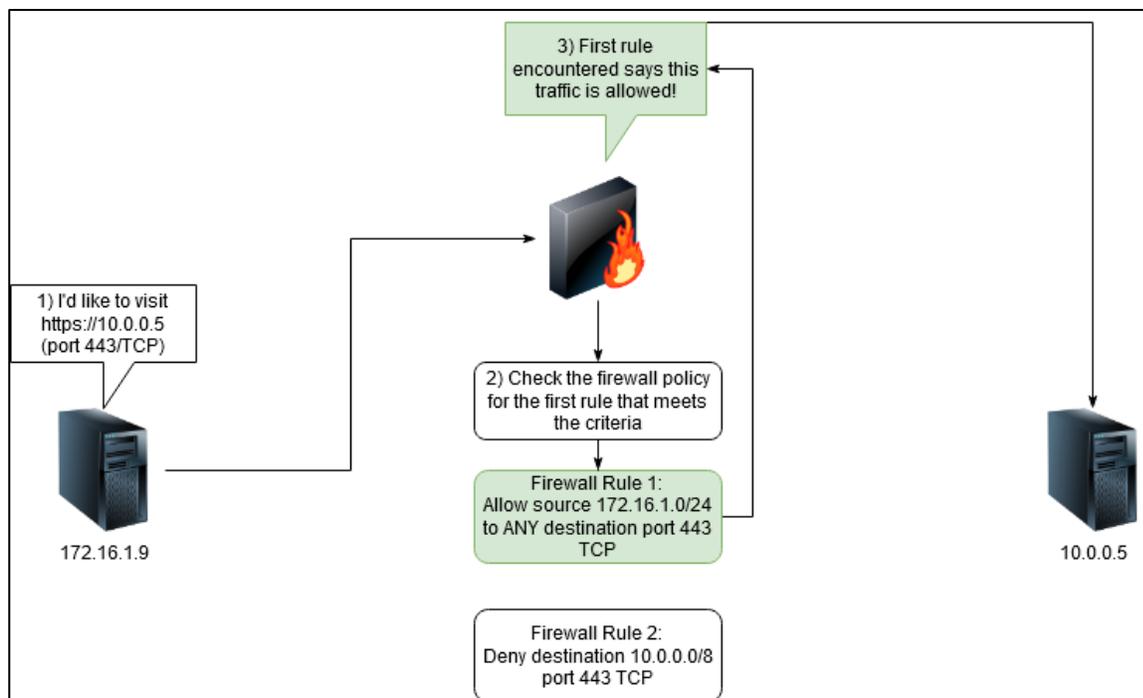
Let's assume the block rule not allowing HTTPS traffic to 10.0.0.0/8 is placed first, THEN the firewall rule allowing HTTPS traffic outbound to any destination is second. 172.16.1.9 wants to

access <https://10.0.0.5> (port 443/TCP). This traffic will be denied, because the deny rule is placed above the allow rule, and so the deny rule is processed first.

What would happen if the rule order swapped, and the allow HTTPS to any destination rule came first? The traffic would be allowed because the firewall sees the HTTPS traffic, sees the allow rule to any destination and no longer cares about the rest of the stack of firewall rules and how they apply. In fact, that deny rule would never apply to HTTPS traffic coming from the 172.16.1.0/24 network. Keep this in mind when creating pfSense firewall policies. Check out the diagrams below to see this process illustrated.



14-13: This diagram shows one instance of how firewall rule order influences whether or not traffic is allowed to flow to a destination. In this instance, we see the deny rule is placed above the allow rule that would allow 172.16.1.9 to access host 10.0.0.5 over port 443 TCP (HTTPS). However, since the deny rule is placed above the allow rule, and matches the criteria, the traffic is denied.



14-14: This diagram, the counterpart to *fig. 14-13*, illustrates how network traffic is affected by rule order. In this instance, a firewall rule allowing traffic from 172.16.1.0/24 to any destination over port 443/TCP is placed above the deny rule from *fig. 14-13*. In this instance, the traffic is allowed. In fact, firewall rule 2 will never be triggered so long as the source IP address is in the 172.16.1.0/24 network.

Now, let's talk about *implicit deny any*, sometimes known as *default deny any*. As the name implies, if there is no firewall rule defined to explicitly allow a certain kind of traffic, then the firewall will default to not allowing the traffic. Most professional network firewalls operate in this manner. A simple example might be a system requesting access to port 3389/TCP outbound (Microsoft remote desktop protocol). If there is no firewall rule to allow this traffic outbound, the firewall defaults to not allowing it.

No rules are currently defined for this interface
All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule.

14-15: If there are no firewall rules defined on an interface, pfSense is quick to remind users that the default deny any rule will make sure all traffic will be blocked until allow rules are added. Default deny any is also called implicit deny any. There is no rule on the firewall policy pages for this default deny behavior – It exists without having a firewall rule explicitly defined for it.

14.4.2 Firewall Aliases

pfSense has a firewall feature called *Aliases*. Aliases allow users to cluster a group IP addresses, network ranges, ports, and URLs under a single name. If you're a programmer, or familiar with programming concepts, think of an alias as an array that holds a set of values. Let's use an example and say someone had a group of SQL servers – 172.16.1.12, 172.16.1.13, and 172.16.1.14. The database administrators are on network 192.168.77.0/24. The DBAs want access to the SQL servers on port 3306/TCP, and the firewall admin needs to create rules to allow this traffic. We could create two aliases here: A network alias to define 192.168.77.0/24 as *DBA_Net*, and an IP address alias for our three SQL servers named *SQL_Servers*. That way, instead of being required to create the same rule 3 times to grant the DBAs access to all three servers, only one rule would be required. Not only that, If the DBA subnet ever changes, or the number of SQL servers increased, the aliases could be modified to address that.

Name	Values	Description
DBA_Net	192.168.77.0/24	Network segment where the DBAs frolic in the SQL forest.
SQL_Servers	172.16.1.12, 172.16.1.13, 172.16.1.14	A collection of mysql servers

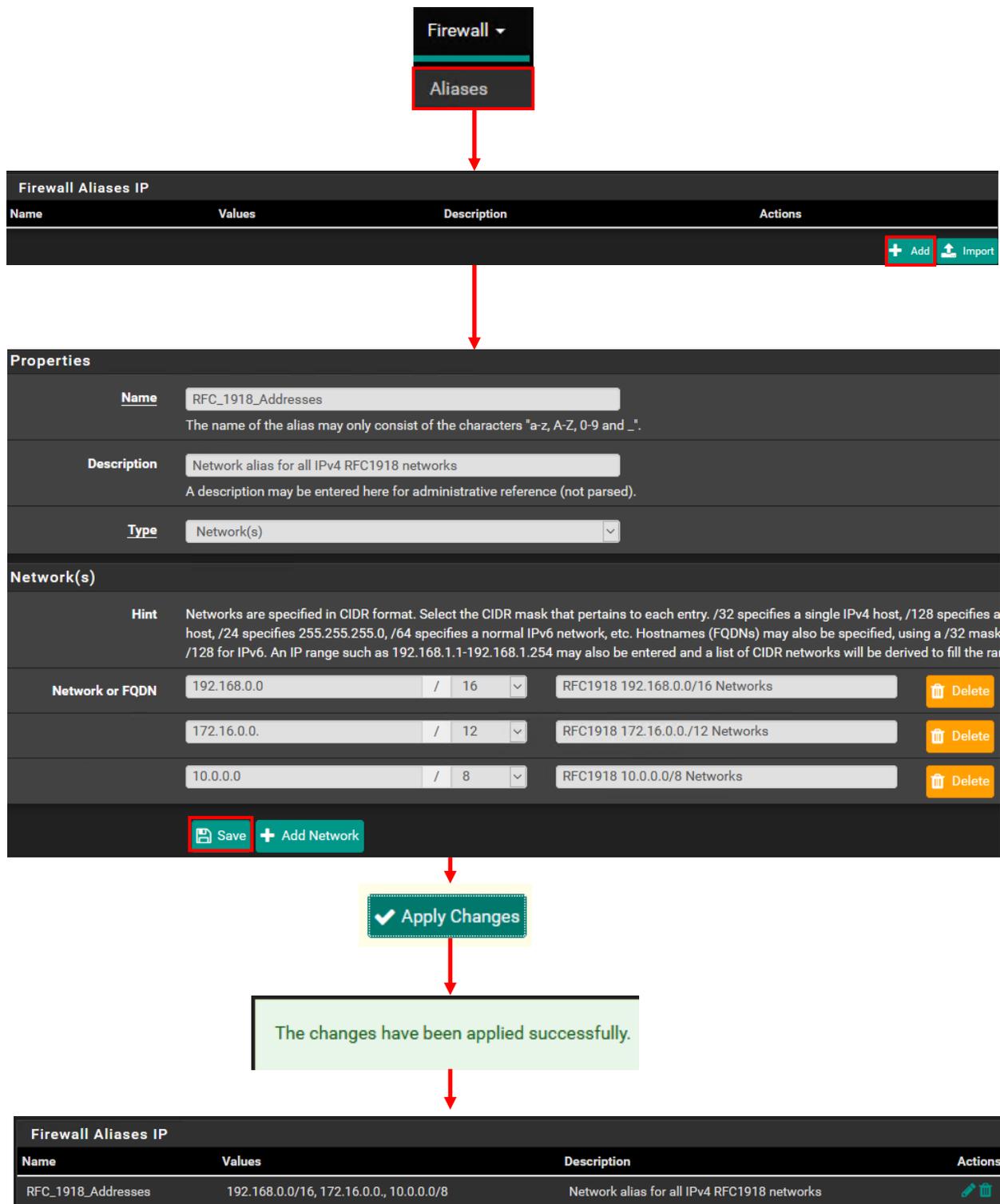
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
0 / 0 B	IPv4 TCP	DBA_Net	*	SQL_Servers	3306	*	none		Access to the SQL servers

14-16: Defining the aliases *DBA_Net* and *SQL_Servers* (1) lets us create firewall rules that reference to multiple IP addresses and networks at once (2).

Students will create an alias named *RFC_1918_Addresses*. To do so, navigate to *Firewall > Aliases* on the pfSense navigation menu. The Aliases page defaults to displaying a tab labeled *IP*. The *IP* tab is used for making and displaying existing IP and network address aliases. Click the *Add* button in the lower left under the *Firewall Aliases IP* window. When the *Edit* page loads, there are host of fields to modify. Focus on the fields in the *Properties* window first. Enter *RFC_1918_Addresses* in the input box of the *Name* field. For the *Description* field, enter *Network alias for all IPv4 RFC1918 networks*. In the *Type* field, select *Network(s)* from the drop-down. Under the *Network(s)* window, click the *Add Network* button twice. Students should have three columns, with six input boxes in total in the *Network(s)* window – Three with the text *Address* in the input box, and three with the word *Description* in their input box. Enter the following information:

Address	Description
192.168.0.0/16	RFC1918 192.16.0.0/16 Networks
172.16.0.0/12	RFC1918 172.16.0.0/12 Networks
10.0.0.0/8	RFC1918 10.0.0.0/8 Networks

Once finished, click *Save*, then *Apply Changes*. The Firewall Aliases IP window changes to reflect information about the newly created alias.



14-17: To create the *RFC_1918_Addresses* firewall alias, Navigate to Firewall > Aliases. On the Firewall Aliases IP screen, click the Add button. Enter the *RFC_1918_Addresses* as the name, a fitting description for the alias, and set the *Type* to *network(s)*. In the *Network(s)* section, click the *Add Network* button twice, then enter the networks and descriptions specified in the figure above. Once finished, click the *Save* button, Followed by *Apply Changes*.

14.4.3 Creating Firewall Rules

In this section, students will walk through the process of creating a new firewall rule. In our example firewall rule, we're going to allow access to port 53/UDP (DNS queries) from 172.16.1.0/24 to 172.16.1.1 (LAN network to LAN Interface). Afterwards, students will be provided with a series of screen captures that contain firewall policies for the LAN, WAN and OPT1 interface with slight variations in the firewall policies depending on whether students are using a hosted or bare-metal hypervisor. Students will also be provided with instructions for creating a more secure anti-lockout rule for accessing the web configurator on both bare-metal and hosted hypervisors, allowing them to delete the default anti-lockout rule.

To get started, navigate to *Firewall > Rules* in the pfSense navigation menu. The Firewall rules page defaults to displaying rules for the *WAN* interface. Click the tab labeled *LAN* to bring up rules for the *LAN* interface. By default, the *LAN* interface has 3 allow rules defined in the window labeled *Rules (Drag to Change Order)*: The Anti-Lockout Rule, allowing any IP address to access the pfSense firewall via port 80/TCP or 443/TCP on the *LAN* interface, and two rules allow any IPv4 or IPv6 traffic outbound from the *LAN* network. As the name of the window states, rules can be dragged and dropped by hovering over a column containing a rule they wish to move, and holding down left-click while dragging with the mouse. Any rule that has a checkbox can be dragged and dropped to change the order its processed in. If the rule order is changed at all, users will need to click the Save button below the Rules window, then *Apply Changes* for that rule order change to take effect. Notice that the anti-lock rule has no checkbox, and cannot be re-arranged.

Click either of the *Add* buttons in the lower left portion of the Rules window. Clicking the *Add* with the upward pointing arrow places the rule we're creating at the top of the rule stack. Clicking the *Add* button with the downward facing arrow places our new firewall at the bottom of the rule stack. It doesn't matter which one you pick, because the new rule can be dragged and dropped into position as needed. *The Edit page appears.* There are a variety of settings and windows on this page that all govern various aspects on how the new firewall rule will work. Let's start with the settings in the *Edit Firewall Rule* window.

The *Action* field determines what will happen when this rule is matched. The drop-down options are *Pass* (allow the traffic to occur), *Block* (silently drop the packets) and *Reject* (Send a TCP RST and/or ICMP port unreachable to the send to inform the sender that the connection is prohibited). The *Disable this rule* checkbox allows users to define the firewall rule, add it to the stack of rules on the given interface, but then ensure it does not actually process any traffic. This may be useful for creating a rule that acts as template through the copy option (the overlapping squares) under the *Actions* column on the firewall rules page.

The *Interface* field defines which firewall interface the rule will be applied to. This field usually defaults to the interface whose firewall rules were being viewed before clicking the *Add* button. For example, students were on the firewall rules tab for the *LAN* interface, so the *LAN* interface

is selected. The *Address Family* field is a drop-down that allows users to select what version of the Internet Protocol they want to apply the firewall rule to – *IPv4*, *IPv6*, or both (*IPv4+IPv6*). That leaves the *Protocol* field to define what type of transport protocol this rule should apply to. There are a wide variety of options here, but students will mostly be interested in *TCP*, *UDP*, *TCP/UDP*, *ICMP*, or *Any*.

The *Source* window allows users to define what source IP address, network or built-in network aliases the rule will apply to. The *Source* field features an input box, and a drop-down. By default, the drop-down is set to *any*. If the drop-down is set to *Network*, or *Single host or alias*, the input box next to the drop-down becomes available for students to enter an IP address, alias, or network address. The checkbox labeled *Invert match* allows students to apply inversion (e.g., "apply this rule to everything but what I entered into this input box and/or drop-down"). The *Display Advanced* button allows source port numbers for a firewall rule to be defined. The text beneath the button states that in most cases, users do not want to specify a source port range, but makes the option available for more complex firewall rules as needed.

The *Destination* window, and *Destination* field work almost identically to the *Source* window and field. Users can use pfSense built-in aliases, specify custom created aliases, individual IP addresses, or network ranges that the firewall rule should apply to. What's different is the *Destination Port Range* field below. This field only appears if users have selected *TCP*, *UDP*, or *TCP/UDP* as their *Protocol* in the *Edit Firewall Rule* window. There are four settings in this field – two drop-down menus, and two input boxes. The drop-down menus are labeled *From* and *To*, respectively. When clicked on, they reveal a variety of protocols and the port numbers those protocols usually default to. For example, the SSH protocol default is port 22/TCP. Selecting the SSH (22) alias applies the firewall rule to port 22/TCP, greys out the *Custom* field next to *From* and *To*, and sets the *To* drop-down to SSH (22) as well. For custom ports and protocols, users can select (*other*). This sets both the *From* and *To* drop-downs to (*other*), and allows users to set a custom port, or custom range of ports for their firewall rule. For example, the Squid proxy service defaults to port 3128/TCP. Users would select (*other*) in both the *From* and *To* drop-downs, then enter 3128 in both of the *Custom* input boxes.

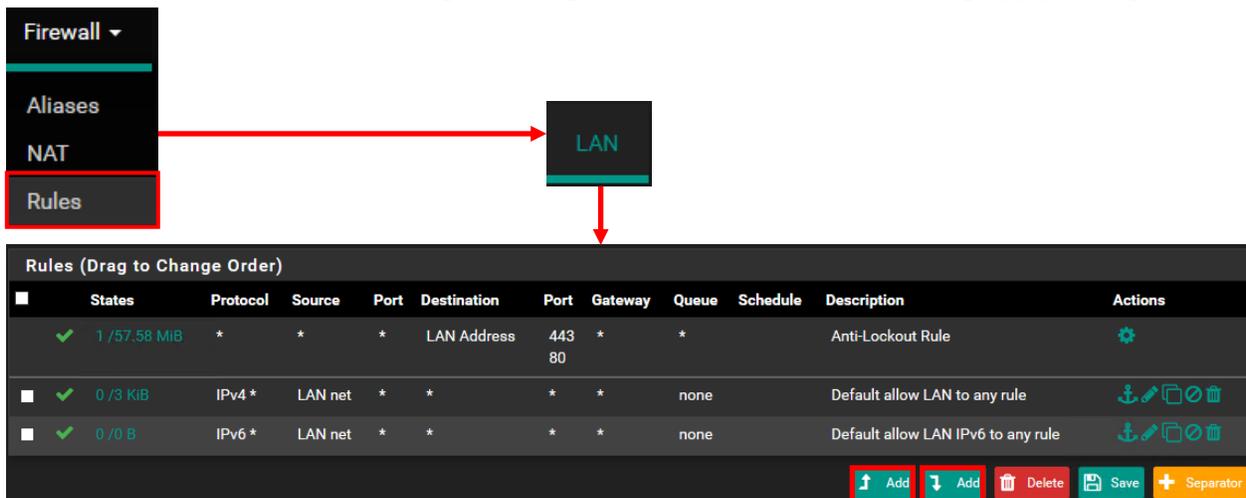
Finally, there is the *Extra Options* window. The *Log* checkbox will log all traffic handled by this rule. pfSense is quick to warn you that if logging is turned on, it will rapidly consume disk space if the firewall handles a lot of traffic, and recommends setting up a remote syslog server for more permanent log storage if desired. Since log space is very limited on the pfSense VM, none of our firewall rules will generate logs. However, this may be a project or lab enhancement for the virtual lab students can experiment with later. The *Description* input box and field allow users to document what the rule affects and why it exists. Its good practice to document the purpose a firewall rule serves, like "SSH access to PCI servers", or "HTTPS access to internet". Make it something short and descriptive. Unfortunately, we won't be covering the advanced options available, but suffice to say, there is a lot of granularity pfSense allows users to apply to their firewall rules if required, or desired.

That covers all of the major fields on the *Edit* page. Students will create a firewall rule with the following settings:

Action: Pass
Interface: LAN
Address Family: IPv4
Protocol: UDP
Source: Network, 172.16.1.0/24
Destination: Single host or Alias, 172.16.1.1
From: DNS (53), To: DNS (53)
Description: Allow DNS traffic to LAN interface

After creating this firewall rule, click *Save* on the bottom of the *Edit* page, then *Apply Changes* on the firewall rules page. Don't worry about the placement of the firewall rule (e.g. top or bottom of the stack) just yet. Take note of the five icons that appear next to each standard rule under the *Actions* column in the *Rules* window: Hovering over each icon pops up a dialogue box that explains what each icon does. Be aware of the edit icon (the pencil) to modify firewall rules as necessary, the copy icon (the overlapping squares) that makes a copy of the currently selected rule (and opens it for editing), and the delete icon (trash bin) for permanently removing rules.

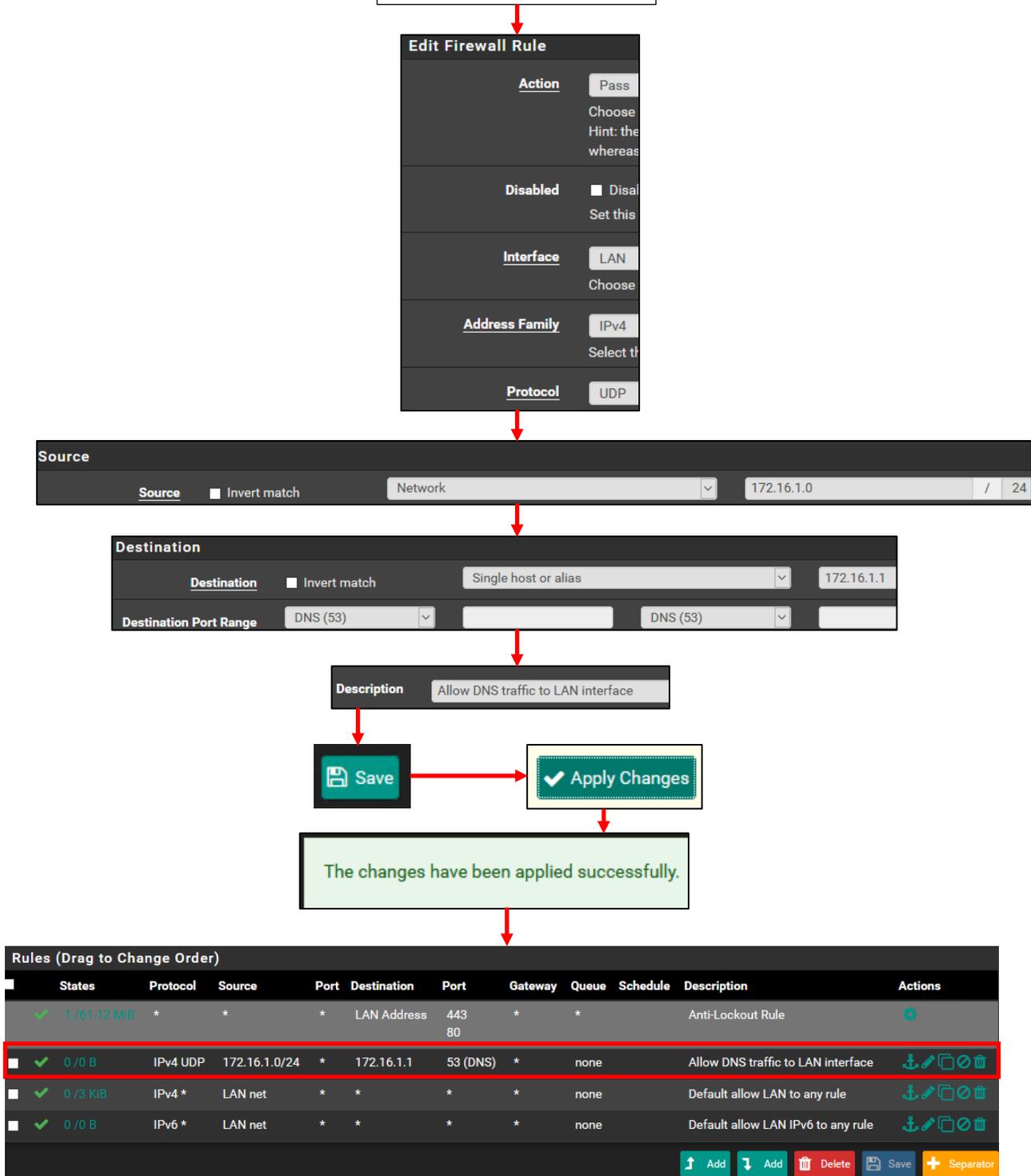
Remember: firewall rules can be dragged and dropped to change the order they are processed in. This is important for the coming sections of this chapter where students will be creating their firewall policies for each of the 3 firewall interfaces. Changing the rule order requires users to the *Save* button below the *Rules (Drag to Change Order)* window, then clicking *Apply Changes*.



Continued to *fig. 14-19*

14-18: To create firewalls via the pfSense webConfigurator, select *Firewall > Rules* from the navigation bar, then select the tab for the interface students wish to create rules on. For our example, click the *LAN* tab. Under the window labeled *Rules (Drag to Change Order)*, click either of the *Add* buttons to begin creating a new firewall rule for the LAN interface. Clicking the *Add* with the upward pointing arrow places the new at the top of the rule stack, while clicking the *Add* button with the downward facing arrow places it at the bottom of the rule stack.

Continued from fig. 14-18



14-19: On the Edit Firewall Rule page, Set the *Action* drop-down to *Pass*, ensure the *Interface* is set to *LAN*, The *Address Family* is set to *IPv4*, and the *Protocol* is set to *UDP*. Under the *Source* section, select *Network* from the drop-down, then enter 172.16.1.0/24 as the network address. In the *Destination* section, select *Single host or alias* from the drop-down, then enter 172.16.1.1. For the *Destination Port Range*, select *DNS (53)* from the drop-down. Finally, enter a description for this firewall rule (e.g., "Allow DNS traffic to LAN interface"), then click the *Save* button. This brings users back to the firewall rules listing for the *LAN* interface, updated with our new rule. Click *Apply Changes* in order to activate the new firewall rule.

14.4.4 Firewall Rule Policy – Hosted Hypervisors

This section will consist of four subsections. Each section will contain a screen capture of the firewall rules on the *WAN*, *LAN* and *OPT1* interface, and a description on why the rules are configured the way they are. The final subsection will provide detailed instructions on disabling the *Anti-Lockout Rule* on the *LAN* interface, and replacing it with something much more secure. **If students are using VirtualBox, VMware Fusion, VMware Workstation, or Microsoft Client Hyper-V (yes, I know it technically isn't a hosted hypervisor), read the next 4 subsections and follow the guidance. Bare-metal hypervisor users: jump to section 14.4.5**

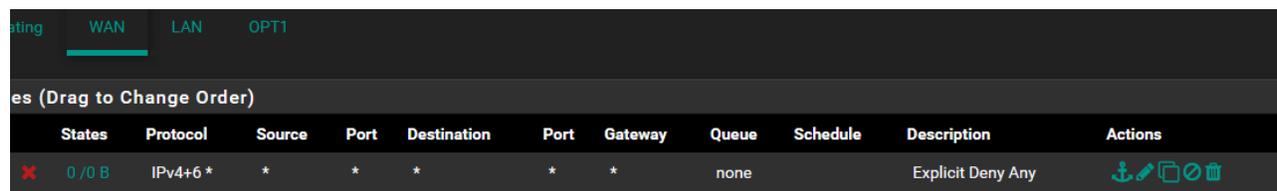
The Rules are the Rules

A couple of quick reminders for the following sections:

With the exception of the *Anti-Lockout Rule* (which we'll be deleting anyway), rules can be dragged and dropped in order to alter what order they are processed in. On the Rules (Drag to Change Order) screen, select the rule to be moved by left-clicking on its column, then left-click again and hold left click to drag the rule to its new location. Afterwards, click the *Save* button.

For students using alternate IP address ranges for the *LAN* (Management) and *OPT1* (IPS1/2) networks, be sure to adjust your firewall rules accordingly.

14.4.4.1 – WAN Interface



States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
X 0 / 0 B	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any	

14-20: The *WAN* interface on hosted hypervisor labs is pretty bare. Technically, hosted hypervisor users don't even need this rule. It's more of a visual reminder that everything inbound on this interface is being blocked.

The *WAN* interface consists of a single block rule deny for all IPv4 and IPv6 traffic using any port or any protocol. This rule exists for students to have a visual reminder of the default deny any rule that exists on this interface. This firewall rule only affects inbound traffic towards the *WAN* interface. This rule will not prevent VMs on the lab networks from accessing physical hosts or the internet. That task is going to fall to the firewall rules on the *LAN* and *OPT1* interfaces, respectively. Remember to click *Apply Changes* when finished.

14.4.4.2 – LAN Interface

Rules (Drag to Change Order)											
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions	
✓ 1 / 771.70 MiB	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule		
✓ 0 / 4 KiB	IPv4 TCP	172.16.1.2	*	172.16.1.1	443 (HTTPS)	*	none		Better Anti-Lockout Rule		
✓ 0 / 0 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	53 (DNS)	*	none		Allow DNS to LAN interface		
✓ 0 / 0 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	123 (NTP)	*	none		Allow NTP to LAN interface		
✓ 0 / 0 B	IPv4 TCP	172.16.1.0/24	*	172.16.1.1	3128	*	none		Allow SQUID to LAN interface		
✓ 0 / 0 B	IPv4 TCP	172.16.1.2	*	172.16.2.2	22 (SSH)	*	none		Allow SSH from hypervisor host to Kali VM		
✗ 0 / 0 B	IPv4+6 *	*	*	RFC_1918_ Addresses	*	*	none		Deny access to RFC1918 addresses		
✓ 0 / 0 B	IPv4 TCP	172.16.1.0/24	*	*	443 (HTTPS)	*	none		Allow HTTPS outbound		
✓ 0 / 0 B	IPv4 UDP	172.16.1.0/24	*	*	123 (NTP)	*	none		Allow NTP outbound		
✗ 0 / 0 B	IPv4+6 *	*	*	*	*	*	none		Explicitly Deny Any		

14-21: Firewall rules for the LAN interface for hosted hypervisors. Notice that the default Anti-Lockout Rule is still present. Section 14.4.4.4 shows students how to remove this rule. **Make absolutely sure the Better Anti-Lockout Rule is immediately under the default Anti-Lockout Rule.**

The first thing most students will notice is the *Anti-Lockout Rule* that pfSense creates by default, followed by a user-created firewall rule named *Better Anti-Lockout Rule*. As the name implies, it's a better, more secure version of the default anti-lockout rule because it only allows a single IP address to access the pfSense webConfigurator over HTTPS (port 443/TCP). Recall that 172.16.1.2 should be the IP address of your hypervisor host's virtual network adapter, while 172.16.1.1 should be the IP address of the pfSense LAN interface. Neither of these addresses should change, because they are both statically assigned, and are outside of the DHCP scope for the LAN network. It is important this rule specifies the correct IP address of the hypervisor host virtual adapter and the LAN interface, because we'll be disabling the standard *Anti-Lockout Rule* in section 14.4.4.4. **It is recommended to always have the Better Anti-Lockout Rule be the first rule at the top of the stack to ensure students do not accidentally lock themselves out of the pfSense firewall.**

The next four rules are all allow rules for various protocols – The first three rules allow VMs on the LAN network to access the LAN interface IP address for DNS, NTP and HTTP proxy services that we set up earlier in this chapter. The fourth rule allows SSH traffic (port 22/TCP) from our hypervisor host's virtual network card (172.16.1.2) to 172.16.2.2 – the Kali VM. Later on, students

will learn how to configure remote access to their lab VMs via the SSH protocol. This rule is to help enable that.

The final four rules are sort of related in function. The first of these rules, is a deny rule for IPv4 and IPv6 traffic from any source IP address to any network in the RFC_1918_Addresses alias we created earlier. Up until this point, the order of our allow rules, even the *Better Anti-Lockout Rule*, technically did not matter. However, **any allow rules that reference any RFC1918 address destinations must be placed above this rule. This is especially important for our Better Anti-Lockout Rule, and rules for accessing network services provided by the pfSense VM (e.g., NTP, DNS, HTTP proxy, etc.).** This rule is used to enforce segmentation of our lab networks from one another, as well as any physical networks the virtual machine lab is connected to.

The next rule allows hosts in the LAN network (172.16.1.0/24) to make outbound connections over port 443/TCP. The placement of this rule after the rule blocking access to all RFC1918 networks is 100% intentional. This rule combo will allow our LAN VMs access to internet resources over the HTTPS protocol, but will prevent lab VMs from being able to access the webconfigurator, or any other local network resources over HTTPS. Combined with the squid proxy service the pfSense VM provides, LAN VMs should have access to download internet resources over HTTP or HTTPS with no problems, but should not be able to touch web servers on the other lab network segments, or the local physical network. The next firewall rule is to allow NTP access to any external IP address, and operates identically to the HTTPS rule above: allow virtual machines to access any NTP server that is not an RFC1918 address (except the pfSense VM).

Our final rule is entirely optional and is a physical reminder of the implicit deny any rule, that blocks any traffic without a pass rule to specifically allow it. **As a final reminder, the order of these firewall rules is extremely important. Refer to fig. 14-21 for the correct order to place the LAN interface firewall rules in.** If students were required to drag and drop rules into place, remember to click *Save*. When all rules have been created and placed in the proper order, click *Apply Changes*.

What's the time: Why are there two NTP firewall rules?

Upon examining *fig. 14-21*, some students may be asking themselves why there are two firewall rules for NTP traffic. Most operating system installers will attempt to synchronize to an NTP server of some sort during the installation process in order to set the time. The problem is that some installers insist upon using a particular NTP source instead of asking the user what server they should use.

If access to the NTP server the installer wants to sync against is blocked, usually the worst that happens is that the installer uses the BIOS (hardware) clock to set the time. The OS installer will be delayed for a moment or two, then continue as normal. In other, more extreme cases, I've seen the installer fail catastrophically because the system clock was too far out of sync to trust the software package sources used to install operating system components. To avoid this, the LAN network has two firewall rules for NTP: one to allow access to the NTP server on the pfSense firewall, and a second so that VMs can reach their preferred NTP server during operating system installation.

14.4.4.3 OPT1 Interface

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✗ 0 / 0 B	IPv4 *	172.16.2.3	*	*	*	*	none		Deny Metasploitable 2 outbound	
✓ 0 / 0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	53 (DNS)	*	none		Allow DNS traffic to OPT1 interface	
✓ 0 / 0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	123 (NTP)	*	none		Allow NTP traffic to OPT1 interface	
✓ 0 / 0 B	IPv4 TCP	172.16.2.0/24	*	172.16.2.1	3128	*	none		Allow SQUID proxy traffic to OPT1 interface	
✗ 0 / 1 KiB	IPv4+6 *	*	*	RFC_1918_ Addresses	*	*	none		Deny access to all RFC1918 addresses	
✗ 0 / 0 B	IPv4+6 TCP	*	*	*	80 (HTTP)	*	none		Deny HTTP outbound (force proxy use)	
✗ 0 / 0 B	IPv4+6 TCP	*	*	*	21 (FTP)	*	none		Deny FTP outbound (force proxy use)	
✓ 0 / 0 B	IPv4+6 *	*	*	*	*	*	none		Allow ANY rule (custom malware/C2)	

14-22: The *OPT1* firewall rules are designed to allow pfSense to provide network services, prevent the Metasploitable 2 VM from being able to communicate outside of the network segment, and enable malware analysis.

The idea behind the *OPT1* network is to enable this network segment for malware analysis purposes. The very first rule is a deny rule explicitly denying access from 172.16.2.3 to any external networks. **This is extremely important because Metasploitable 2 is an intentionally**

vulnerable virtual machine. Allowing it to access external resources, or external systems to access it could present a major security risk. Make sure that this is the first rule on the OPT1 firewall rule stack. Later on, if students acquire additional vulnerable virtual machines for expanding their lab environment (e.g., "boot2root" VMs), they should be assigned static IP addresses, and additional firewall rules should be created (placed at the top of the OPT1 firewall rule stack) to deny them access to external resources as well.

The next four rules are nearly identical to their counterparts on the LAN interface: Allow rules for other hosts on the *OPT1* networks to utilize pfSense *OPT1* interface for DNS, NTP, and HTTP proxy services, followed by a rule to deny hosts on this network access to any RFC1918 addresses. The two deny rules immediately following the RFC1918 block rule specifically deny access to ports 80 and 21 outbound – these are the HTTP and FTP protocols. The Description states that this is to force all hosts that want external FTP or HTTP access to utilize the SQUID proxy service on 172.16.2.1, the *OPT1* interface. That brings us to the final rule, the Allow any/any outbound.

Malware analysis is something of a complex topic. Creating a virtual lab and network that is permissive enough to allow external network access but prevent malware from attacking other local network segments is a very complex task. The way these rules are setup mean that any virtual machines created for the purpose of detonating malware will have access to vital network service protocols for name resolution and time-keeping, while plaintext protocols like FTP and HTTP are able to be logged through the SQUID proxy service on the pfSense VM. We don't have logging or log forwarding to a syslog server enabled for the pfSense SQUID proxy service, but the option is available as a possible enhancement students can enable for their lab later on. Now, what about other protocols? A lot of malware uses HTTPS for external command and control, or other custom ports. The allow any/any rule allows this traffic outbound. The idea being that the malware analysis/detonation virtual machines are placed on the IPS2 network, and pass through the inline IDS. The IDS logs network connections and lets students see what connections are being made (including HTTP and FTP), and where.

Note: Another possible enhancement to look into would be enabling squid to do HTTPS proxying. This is something of both a complex and contentious topic, but for the purposes of a malware analysis network, enabling HTTPS proxying, and forwarding the squid proxy logs would allow students to see what HTTPS URLs systems are visiting.

What if I Don't plan on Doing Malware Analysis?

As stated above, the *OPT1* network is intentionally set up to be a bit more permissive to allow for malware analysis, and observation of payload delivery and/or command and control (C2). Well, not everyone wants to do malware analysis or is comfortable with full outbound internet access over any port or protocol. An alternative firewall rule set for the *OPT1* interface would be to mirror the firewall rules of the *LAN* interface: Delete the last 3 firewall rules on *figure 14-22* and replace them with a rule allowing HTTPS outbound, NTP outbound, and (optionally) the explicit deny any rule. This limits outbound internet access, just like the *LAN* interface. **Remember to keep the block rule for Metasploitable 2 (and any other additional boot2root VMs) at the very top of your firewall rule stack. I cannot overstate the importance of keeping your intentionally vulnerable virtual machines isolated.**

■	✖	0 / 0 B	IPv4+6	TCP	*	*	*	80 (HTTP)	*	none	Deny HTTP outbound (force proxy use)	
■	✖	0 / 0 B	IPv4+6	TCP	*	*	*	21 (FTP)	*	none	Deny FTP outbound (force proxy use)	
■	✔	0 / 0 B	IPv4+6	*	*	*	*	*	*	none	Allow ANY rule (custom malware/C2)	

14-23: Replace these bottom three rules...

■	✔	0 / 0 B	IPv4	TCP	172.16.2.0/24	*	*	443 (HTTPS)	*	none	Allow HTTPS traffic outbound	
■	✔	0 / 5 KIB	IPv4	UDP	172.16.2.0/24	*	*	123 (NTP)	*	none	Allow NTP traffic outbound	
■	✖	0 / 93.66 MiB	IPv4+6	*	*	*	*	*	*	none	Explicit Deny Any	

14-24: ...With these three rules to make the *OPT1* network behave like the *LAN* network.

14.4.4.4 Removing the Default Anti-Lockout Rule

Note: This may be jumping the gun a little bit, but if there is any doubt in your mind that you may make a mistake with these instructions, create a VM snapshot prior to performing these steps. Technically, we haven't covered VM snapshots yet, but instructions on how to do this are covered in the hypervisor setup guide for your hypervisor of choice. Check out the following sections:

Microsoft Client Hyper-V: Chapter 9, [section 9.6](#), pp. 189-193

Oracle VirtualBox: Chapter 10, [section 10.8](#), pp. 299-301

VMware Fusion: Chapter 11, [section 11.6](#), pp. 399-402

VMware Workstation: Chapter 12, [section 12.7](#), pp. 507-511

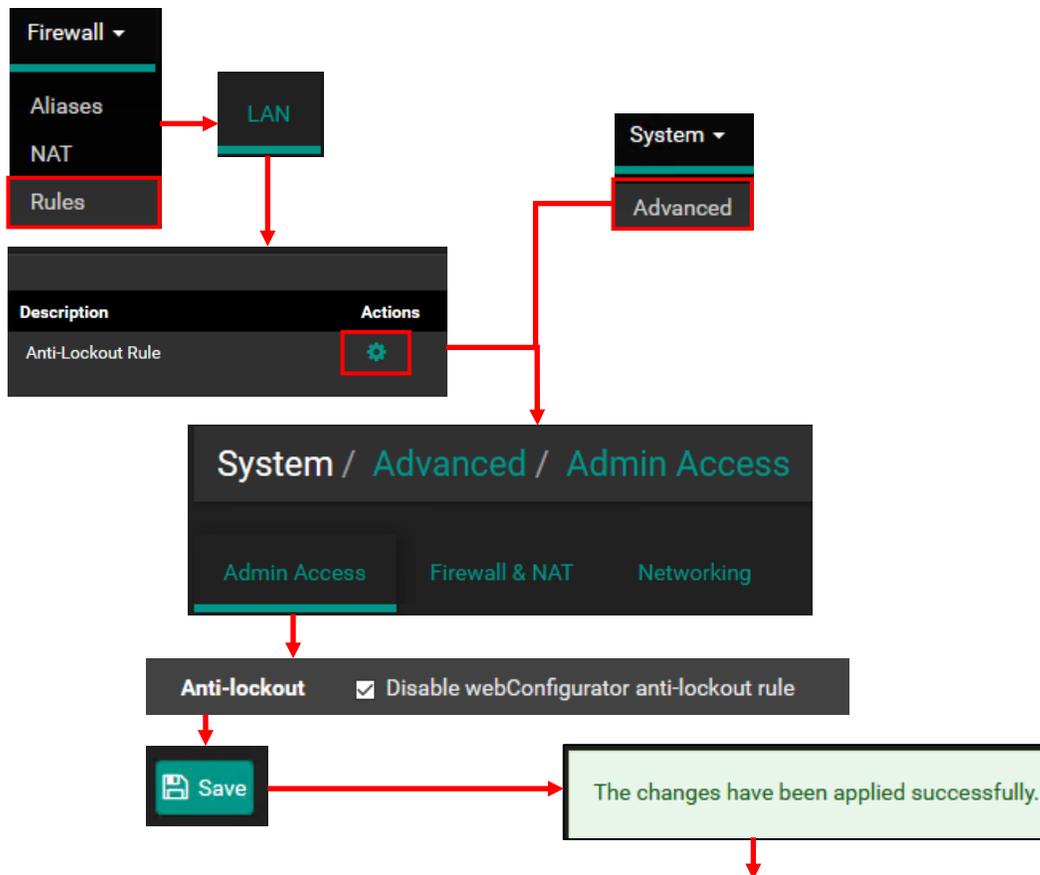
Figure 14-21 depicts the firewall rule stack for the LAN interface. Currently, the default *Anti-Lockout Rule* is enabled, followed by a standard firewall rule with the description, *Better Anti-Lockout Rule*. The purpose of the anti-lockout rule is to make sure that pfSense users do not accidentally lose access to the webConfigurator. By default, this rule allows any IP address on the LAN network to connect to the webConfigurator on ports 80 or 443. pfSense automatically ensures this is the first firewall rule on the LAN interface.

The firewall rule students were instructed to create allows only the hypervisor host's virtual network adapter (172.16.1.2) to access the webConfigurator. **Students must be aware that this rule will not automatically be pushed to the top of the firewall rule stack. They will need to manage that task manually if they modify their firewall rule policy for the LAN interface. After disabling the Anti-Lockout Rule, it is extremely important to make sure the Better Anti-Lockout rule is always the first rule on the LAN interface.** Why go through all the trouble of manually creating a rule when one exists that already solves the problem automatically? Let's talk about the security concepts least privilege, and defense in depth.

Least privilege is the idea of providing only the minimum amount of access necessary to perform a task. For example, limiting the number of systems allowed to access the pfSense webConfigurator is important in order to enhance the security of the lab network. None of the lab virtual machines should have access to the webConfigurator, because it represents a security risk. If one of the lab virtual machines becomes compromised, it could potentially be used to attack the webconfigurator, gain access, and compromise the network segmentation of the entire lab network.

Least privilege is a single layer of defense that can be used to make the pfSense VM more secure, but it's only a single layer. **Students will want to have multiple, overlapping defensive techniques in place to protect the pfSense VM because of how pivotal it is to the function of the lab environment.** That's why they should ensure the pfSense VM is consistently updated, the webConfigurator admin account has strong password, least privilege is being applied, etc. This concept of multiple overlapping layers of defense is called defense in depth.

To disable the default Anti-Lockout Rule, select *Firewall > Rules* in the pfSense navigation menu, then click on the *LAN* tab. Under the *Actions* column, the *Anti-Lockout Rule* has a single gear icon displayed. Click that gear, and students will be redirected to the *Admin Access* tab, under the *System > Advanced* page. This page can also be reached directly via the *System > Advanced* option in the navigation menu. Under the webConfigurator window, locate the field labeled *Anti-lockout*. Click the checkbox labeled *Disable webConfigurator anti-lockout rule*. Scroll to the bottom of the page, and click *Save*. The new settings will be re-applied automatically. After about 20 seconds, the page will automatically refresh. If readers did everything correctly, nothing should happen, and access to the webConfigurator should continue, uninterrupted. If students did not create the *Better Anti-Lockout Rule*, and place it correctly, they will lose access to the webConfigurator. Check out the sidebar *Have you looked under the floormat?* for instructions on how to regain access.



Rules (Drag to Change Order)											
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions	
1 / 35 KIB	IPv4 TCP	172.16.1.2	*	172.16.1.1	443 (HTTPS)	*	none		Better Anti-Lockout Rule		

14-25: The *Admin Access* tab on the *System > Advanced* page can be access directly via the pfSense navigation menu, or by clicking the gear icon in the *Actions* column of the *Anti-Lockout Rule* on the *LAN* interface firewall rules list. Once there, Check *Disable webConfigurator anti-lockout rule*, save your changes, and pray that the session to the webConfigurator doesn't suddenly time out.

Have you looked under the floormat?

If students have managed to lock themselves out of the webConfigurator, open the pfSense command-line menu from console their hosted hypervisor choice, and run the *Set Interface(s) IP address* wizard for the *LAN* interface again, with the same answers supplied previously (e.g., 172.16.1.1/24, no IPv6, no default gateway, DHCP: 172.16.1.10-172.16.1.254, no to webconfigurator over HTTP). Upon completion, pfSense will re-enable the anti-lockout rule, providing access to the webConfigurator once more.

14.4.5 Firewall Rule Policy – Bare-metal Hypervisors

This section is going to consist of five subsections. Each section will contain a screen capture of the firewall rules on the *WAN*, *LAN* and *OPT1* interface, and a description on why the rules are configured the way they are. The fourth subsection will provide detailed instructions on how to remove the default pfSense Anti-Lockout Rule from the *LAN* interface. In the final subsection, students will also learn the process for removing the allow all rule from the *WAN* interface, and replacing it with a manual anti-lockout rule for a management workstation. **This section is specifically for bare-metal hypervisors only (VMware ESXi).**

14.4.5.1 WAN Interface

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 0 / 7.94 MiB	IPv4+6 *	*	*	*	*	*	none		Allow all ipv4+ipv6 via pfSsh.php	
✓ 0 / 0 B	IPv4 TCP	10.0.0.3	*	10.0.0.206	443 (HTTPS)	*	none		Better Anti-Lockout Rule	
✓ 0 / 0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	8000	*	none		Allow Splunk Web access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.3	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to IPS VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.3	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to Kali VM	
✗ 0 / 0 B	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any WAN	

14-26: Firewall rules for the *WAN* interface of a pfSense VM, hosted on a bare-metal hypervisor (VMware ESXi). The *WAN* interface firewall ruleset for bare-metal hypervisors is much more complex, because users will be accessing the webconfigurator, and their lab VMs through the *WAN* interface. Ensure that both your management workstation, and the IP address of the *WAN* interface are either statically configured, or have a static DHCP allocation on your physical network. If students are setting up a lab on a corporate network where client workstations can't get static DHCP allocations but servers (like, say your bare-metal hypervisor server, and the *WAN* interface of the pfSense) can get a static IP address or DHCP allocation, considering setting up a bastion host (jump box) on that network. This process is covered later.

The *WAN* interface firewall rules for pfSense on VMware ESXi (or other bare-metal hypervisors) are going to look much different from the set of rules for the *WAN* interface on hosted hypervisors. This is because students access their lab virtual machines, and the pfSense webConfigurator, through the *WAN* interface of their pfSense VM. The firewall rules below (combined with some static routing, and guidance on setting up remote access – both covered later) will allow enable readers to configure remote access to their virtual machines using the SSH protocol (port 22/TCP), and will also allow access to the Splunk search head (port 8000/TCP) on the SIEM VM much later.

Students will notice that most of the rules for the WAN interface have a source IP address of 10.0.0.3. They may also notice the *Better Anti-Lockout Rule* that specifies a destination IP address of 10.0.0.206. My local physical network uses the subnet 10.0.0.0/24. The IP address of my windows workstation that I use to manage and access my lab environment is 10.0.0.3, while the IP address assigned to the WAN interface of my pfSense VM is 10.0.0.206. **Students will need to substitute these IP addresses with the IP address of their management workstation, and pfSense WAN interface accordingly.**

It is extremely important that your management workstation (and/or optionally, a bastion host – covered in chapter 16) and the WAN interface IP address of your pfSense Virtual Machine both either have a static IP address, or a static DHCP mapping from the device that provides DHCP services for your physical network. This should have been mentioned in the hypervisor setup guide, but because of how important this is to ensure students have consistent access to the webConfigurator and their lab environment, it bears repeating to drive the point home.

The *Better Anti-Lockout Rule* will need to have the IP address of your management workstation as the Source IP address, and the IP address of the WAN interface of your pfSense VM as the destination. Each of the access rules to specific lab VMs will also need to have the IP address of the management workstation as the source address. We haven't yet addressed the rule that allows all traffic on the WAN interface we had to enable from the command line (*Allow all ipv4+ipv6 via pfSsh.php*), but we'll be getting to that in section 14.4.5.5.

The final firewall rule at the very bottom of the stack is labeled *Explicit Deny Any*. This rule exists for students to have a visual reminder of the default deny any rule that exists on this interface. This firewall rule only affects inbound traffic towards the WAN interface. This rule will not prevent VMs on the lab networks from accessing physical hosts or the internet. That task is going to fall to the firewall rules on the LAN and OPT1 interfaces, respectively. Remember, the firewall rules can be dragged and dropped to change their order if necessary, but users will need to click the Save button to apply that rule order. Once finished, click *Apply Changes*.

Note: When and if students decide to create additional VMs for their lab environment, and they want to enable remote access to those VMs, they will need to define firewall rules to allow that network connectivity through the WAN interface. For example, a student sets up a Windows Server virtual machine on the OPT1 network to start emulating an active directory network or some such, and wants to use the RDP protocol to access the VM. They would need to specify a firewall rule on the WAN interface (above the *Explicit Deny Any WAN* rule, if students are using it) from the source IP address of their management workstation (or bastion host, or a host alias, covering multiple source IP addresses) to the destination IP address of the Windows Server VM.

14.4.5.2 LAN Interface

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 0/0 B	*	*	*	LAN Address	443 80	*	*		Anti-Lockout Rule	⚙️
✓ 0/0 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	53 (DNS)	*	none		Allow DNS traffic to LAN interface	📌🔗🗑️
✓ 0/0 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	123 (NTP)	*	none		Allow NTP traffic to LAN interface	📌🔗🗑️
✓ 0/0 B	IPv4 TCP	172.16.1.0/24	*	172.16.1.1	3128	*	none		Allow Squid proxy traffic to LAN interface	📌🔗🗑️
✗ 0/0 B	IPv4+6	*	*	RFC_1918_Addresses	*	*	none		Deny access to all RFC1918 addresses	📌🔗🗑️
✓ 0/0 B	IPv4 TCP	172.16.1.0/24	*	*	443 (HTTPS)	*	none		Allow HTTPS traffic outbound	📌🔗🗑️
✓ 0/0 B	IPv4 UDP	172.16.1.0/24	*	*	123 (NTP)	*	none		Allow NTP traffic outbound	📌🔗🗑️
✗ 0/0 B	IPv4+6	*	*	*	*	*	none		Explicit Deny Any LAN	📌🔗🗑️

14-27: Firewall rules for the LAN interface for bare-metal hypervisors. This ruleset is similar to the rules used on the LAN interface for hosted hypervisors.

The first rule at the top of our firewall rule stack on the LAN interface is the default *Anti-Lockout Rule* pfSense provides. We won't be using this rule at all, and will be disabling it momentarily. The next three rules are all allow rules for various protocols, allowing VMs on the LAN network to access the LAN interface IP address for DNS, NTP and HTTP proxy services that we set up earlier in this chapter.

The final four rules are sort of related in function. The first of these rules, is a deny rule for IPv4 and IPv6 traffic from any source IP address to any network in the *RFC_1918_Addresses* alias we created earlier. Up until this point, the order of the allow rules technically did not matter. However, **any allow rules that reference any RFC1918 address destinations must be placed above this rule. This is especially important for accessing network services provided by the pfSense VM (e.g., NTP, DNS, HTTP proxy, etc.).** This rule is used to enforce segmentation of our lab networks from one another, as well as any physical networks the virtual machine lab is connected to.

The next rule allows hosts in the LAN network (172.16.1.0/24) to make outbound connections over port 443/TCP. The placement of this rule after the rule blocking access to all RFC1918 networks is 100% intentional. This rule combo will allow our LAN VMs access to internet resources over the HTTPS protocol, but will prevent lab VMs from being able to access the webconfigurator, or any other local network resources over HTTPS. Combined with the squid proxy service the pfSense VM provides, LAN VMs should have access to download internet resources over HTTP or HTTPS with no problems, but should not be able to touch web servers on the other lab network segments, or the local physical network. The next firewall rule is to allow NTP access to any external IP address, and operates identically to the HTTPS rule above: allow virtual machines to access any NTP server that is not an RFC1918 address.

Our final rule is entirely optional and is a physical reminder of the implicit deny any rule, that blocks any traffic without a pass rule to specifically allow it. **As a final reminder, the order of these firewall rules is extremely important. Refer to fig. 14-27 for the correct order to place the LAN interface firewall rules in.** If students were required to drag and drop rules into place, remember to click *Save*. When all rules have been created and placed in the proper order, click *Apply Changes*.

What's the time: Why are there two NTP firewall rules?

Upon examining *fig. 14-27*, some students may be asking themselves why there are two firewall rules for NTP traffic. Most operating system installers will attempt to synchronize to an NTP server of some sort during the installation process in order to set the time. The problem is that some installers insist upon using a particular NTP source instead of asking the user what server they should use.

If access to the NTP server the installer wants to sync against is blocked, usually the worst that happens is that the installer uses the BIOS (hardware) clock to set the time. The OS installer will be delayed for a moment or two, then continue as normal. In other, more extreme cases, I've seen the installer fail catastrophically because the system clock was too far out of sync to trust the software package sources used to install operating system components. To avoid this, the LAN network has two firewall rules for NTP: one to allow access to the NTP server on the pfSense firewall, and a second so that VMs can reach their preferred NTP server during operating system installation.

14.4.5.3 OPT1 Interface

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✗ 0/0 B	IPv4 *	172.16.2.3	*	*	*	*	none		Deny Metasploitable 2 outbound	📌✎🗑️🔍
✓ 0/0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	53 (DNS)	*	none		Allow DNS traffic to OPT1 interface	📌✎🗑️🔍
✓ 0/0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	123 (NTP)	*	none		Allow NTP traffic to OPT1 interface	📌✎🗑️🔍
✓ 0/0 B	IPv4 TCP	172.16.2.0/24	*	172.16.2.1	3128	*	none		Allow Squid proxy traffic to OPT1 interface	📌✎🗑️🔍
✗ 0/0 B	IPv4+6 *	*	*	RFC_1918_Addresses	*	*	none		Deny access to all RFC1918 addresses	📌✎🗑️🔍
✗ 0/0 B	IPv4+6 TCP	*	*	*	80 (HTTP)	*	none		Deny HTTP outbound (force proxy use)	📌✎🗑️🔍
✗ 0/0 B	IPv4+6 TCP	*	*	*	21 (FTP)	*	none		Deny FTP outbound (force proxy use)	📌✎🗑️🔍
✓ 0/0 B	IPv4+6 *	*	*	*	*	*	none		Allow ANY rule (custom malware/C2)	📌✎🗑️🔍

14-28: Getting de ja vu? This ruleset for the *OPT1* interface is identical to the one we set up on hosted hypervisors.

The idea behind the *OPT1* network is to enable this network segment for malware analysis purposes. The very first rule is a deny rule explicitly denying access from 172.16.2.3 to any external networks. **This is extremely important because Metasploitable 2 is an intentionally vulnerable virtual machine. Allowing it to access external resources, or external systems to access it could present a major risk.** Make sure that this is the first rule on the *OPT1* firewall rule stack. Later on, if students acquire additional vulnerable virtual machines for expanding their lab environment (e.g., "boot2root" VMs), they should be assigned static IP addresses, and additional firewall rules should be created (placed at the top of the *OPT1* firewall rule stack) to deny them access to external resources as well.

The next four rules are nearly identical to their counterparts on the LAN interface: Allow rules for other hosts on the *OPT1* networks to utilize pfSense for DNS, NTP, and HTTP proxy services, followed by a rule to deny hosts on this network access to any RFC1918 addresses. The two deny rules immediately following the RFC1918 block rule specifically deny access to ports 80 and 21 outbound – these are the HTTP and FTP protocols. The Description states that this is to force all hosts that want external FTP or HTTP access to utilize the SQUID proxy service on 172.16.2.1. That brings us to the final rule, the Allow any/any outbound.

Malware analysis is something of a complex topic. Creating a virtual lab and network that is permissive enough to allow external network access but prevent malware from attacking other local network segments is a very complex task. The way these rules are setup mean that any virtual machines created for the purpose of detonating malware will have access to vital network service protocols for name resolution and time-keeping, while plaintext protocols like FTP and HTTP are able to be logged through the SQUID proxy service on the pfSense VM. We don't have logging or log forwarding to a syslog server enabled for the pfSense SQUID proxy service, but the option is available as a possible enhancement students can enable for their lab later on. Now, what about other protocols? A lot of malware uses HTTPS for external command and control, or other custom ports. The allow any/any rule allows this traffic outbound. The idea being that the malware analysis/detonation virtual machines are placed on the IPS2 network, and pass through the inline IDS. The IDS logs network connections and let's students see what connections are being made (including HTTP and FTP), and where.

Note: Another possible enhancement to look into would be enabling squid to do HTTPS proxying. This is something of both a complex and contentious topic, but for the purposes of a malware analysis network, enabling HTTPS proxying, and forwarding the squid proxy logs would allow students to see what HTTPS URLs systems are visiting.

What if I Don't plan on Doing Malware Analysis?

The *OPT1* network is intentionally set up to be a bit more permissive to allow for malware analysis, and observation of payload delivery and/or command and control (C2). Well, not everyone wants to do malware analysis or is comfortable with full outbound internet access over any port or protocol. An alternative firewall rule set for the *OPT1* interface would be to mirror the firewall rules of the LAN interface: Delete the last 3 firewall rules on figure 14-28 and replace them with a rule allowing HTTPS outbound, NTP outbound, and (optionally) the explicit deny any rule. This limits outbound internet access, just like the *LAN* interface. **Remember to keep the block rule for Metasploitable 2 (and any other additional boot2root VMs) at the very top of your firewall rule stack. I cannot overstate the importance of keeping your intentionally vulnerable virtual machines isolated.**

■	✗	0 / 0 B	IPv4+6	TCP	*	*	*	80 (HTTP)	*	none	Deny HTTP outbound (force proxy use)	
■	✗	0 / 0 B	IPv4+6	TCP	*	*	*	21 (FTP)	*	none	Deny FTP outbound (force proxy use)	
■	✓	0 / 0 B	IPv4+6	*	*	*	*	*	*	none	Allow ANY rule (custom malware/C2)	

14-29: Replace these bottom three rules...

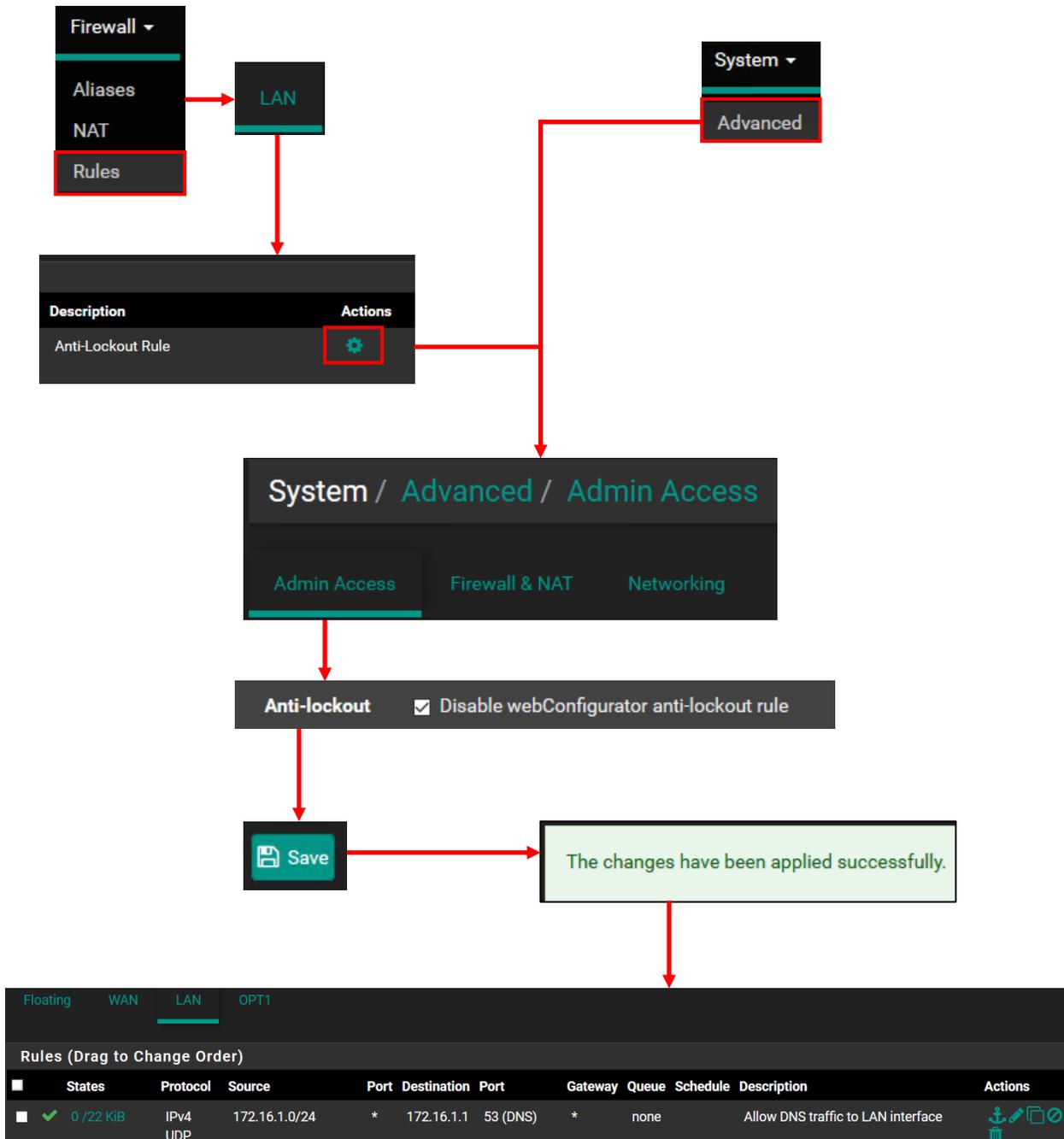
■	✓	0 / 0 B	IPv4	TCP	172.16.2.0/24	*	*	443 (HTTPS)	*	none	Allow HTTPS traffic outbound	
■	✓	0 / 5 KIB	IPv4	UDP	172.16.2.0/24	*	*	123 (NTP)	*	none	Allow NTP traffic outbound	
■	✗	0 / 93.66 MiB	IPv4+6	*	*	*	*	*	*	none	Explicit Deny Any	

14-30: ...With these three rules to make the *OPT1* network behave like the *LAN* network.

14.4.5.4 Removing the Default Anti-Lockout Rule

Figure 14-27 depicts the firewall rule stack for the *LAN* interface. Currently, the default *Anti-Lockout Rule* is enabled. The purpose of the anti-lockout rule is to make sure that pfSense users do not accidentally lose access webConfigurator. By default, this rule allows any IP address on the LAN network to connect to the webConfigurator on ports 80 or 443. pfSense automatically ensures this is the first firewall rule on the LAN interface. As we discussed in section 14.4.5.1, the lab virtual machines should not be allowed to access the pfSense webConfigurator. Only the management workstation (or a bastion host) should be allowed to reach the webConfigurator or remotely access lab virtual machines, so we're going to disable the *Anti-Lockout Rule* to limit that access.

In the pfSense navigation menu, select *Firewall > Rules*, and click on the *LAN* tab. Under the *Actions* column, the *Anti-Lockout Rule* has a single gear icon displayed. Click that gear, and students will be redirected to the *Admin Access* tab, under the *System > Advanced* page. This page can also be reached directly via the *System > Advanced* option in the navigation menu. Under the webConfigurator window, locate the field labeled *Anti-lockout*. Click the checkbox labeled *Disable webConfigurator anti-lockout rule*. Scroll to the bottom of the page, and click *Save*. The new settings will be re-applied automatically. After about 20 seconds, the page will automatically refresh. If readers did everything correctly, nothing should happen, and access to the webConfigurator should continue uninterrupted.



14-31: The *Admin Access* tab on the *System > Advanced* page can be access directly via the pfSense navigation menu, or by clicking the gear icon in the Actions column of the *Anti-Lockout Rule* on the *LAN* interface firewall rules list. Once there, Check the option to disable the anti-lockout rule and save your changes. This shouldn't affect the current session to the webConfigurator at all.

14.4.5.5 Removing the allow all pfSsh.php firewall rule

Note: This may be jumping the gun a little bit, but if there is any doubt in your mind that you may make a mistake with these instructions, create a VM snapshot prior to performing these steps. Technically, we haven't covered VM snapshots yet, but instructions on how to do this are covered in the VMware ESXi hypervisor setup guide in [section 13.8](#), pp. 658-662. If you elected to use another bare-metal hypervisor, please consult the documentation on how to create virtual machine snapshots for that hypervisor.

Figure 14-26 depicts the firewall rule stack for the WAN interface. Currently, the rule *Allow all ipv4+ipv6 via pfSsh.php* is enabled, followed by a standard firewall rule with the description, *Better Anti-Lockout Rule*. Recall in the ESXi hypervisor setup guide that students were required to run the command `pfSsh.php playback enableallowallwan` to gain our initial access to the webConfigurator. This rule allows any traffic to the WAN interface – and subsequently, the rest of our lab environment. It's very important that the rule is removed, and that the *Better Anti-Lockout Rule* specifies the correct source and destination address.

The firewall rule students were instructed to create allows only their management workstation to access the webConfigurator. It's recommended to make the *Better Anti-Lockout Rule* the first rule at the top of the WAN firewall policy. **Students must be aware that this rule will not automatically be pushed to the top of the firewall rule stack. They will need to manage that task manually if they modify their firewall rule policy for the WAN interface.** Why go through all the trouble of manually creating a rule to only allow a single system access to the lab environment and webconfigurator? Let's talk about the security concepts least privilege, and defense in depth.

Least privilege is the idea of providing only the minimum amount of access necessary to perform a task. Limiting the number of systems allowed to access the pfSense webConfigurator (and the lab network VMs in general) is important in order to enhance the security of the lab network. This is the reason why we disabled the default Anti-Lockout Rule on the LAN interface – none of the virtual machines need access to the webConfigurator, and they don't get that access because it represents a security risk. If one of the lab virtual machines becomes compromised, it could potentially be used to attack the webconfigurator, gain access, and compromise the network segmentation of the entire lab network. Likewise, we will disable the allow all rule to prevent hosts on the physical network from compromising the lab network or pfSense webconfigurator.

Least privilege is a single layer of defense that can be used to make the pfSense VM more secure, but it's only a single layer. **Students will want to have multiple, overlapping defensive techniques in place to protect the pfSense VM because of how pivotal it is to the function of the lab environment.** That's why they should ensure the pfSense VM is consistently updated, the webConfigurator admin account has strong password, least privilege is being applied, etc. This concept of multiple overlapping layers of defense is called defense in depth.

To remove the *Allow all ipv4+ipv6* rule, select *Firewall > Rules* from the pfSense navigation menu, then click on the *WAN* tab. Under the *Actions* column, the *Allow all ipv4+ipv6 via pfSsh.php* rule has a series of icons displayed. Click the trash bin icon in the *Actions* column to delete the rule. A dialogue box pops up asking to confirm if students really wish to delete the rule. Click *OK*, then click *Apply Changes* to save the new firewall rule configuration. If students did not create the *Better Anti-Lockout Rule*, place it correctly (directly under the *pfSsh.php* rule), and define the correct IP addresses, they will lose access to the webConfigurator. Check out the sidebar conversation, *Have you looked in the mailbox?* for instructions on how to regain access.

The image illustrates the process of deleting a firewall rule in pfSense. It consists of three sequential screenshots:

- Initial State:** The 'Rules' table on the WAN interface shows two rules:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
3 / 32.72 MiB	IPv4+6 *	*	*	*	*	*	none		Allow all ipv4+ipv6 via pfSsh.php	[Icons]
0 / 0 B	IPv4 TCP	10.0.0.3	*	10.0.0.206	443 (HTTPS)	*	none		Better Anti-Lockout Rule	[Icons]
- Confirmation:** A dialog box asks: "Are you sure you wish to delete this rule?" with "OK" and "Cancel" buttons.
- Final State:** The 'Rules' table shows the 'Better Anti-Lockout Rule' as the first rule:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 0 B	IPv4 TCP	10.0.0.3	*	10.0.0.206	443 (HTTPS)	*	none		Better Anti-Lockout Rule	[Icons]

14-32: Verify the *Better Anti-Lockout Rule* has been created with the correct source IP, destination IP, port, and protocols specified, and that it is the second rule from the top of the firewall rule policy on the *WAN* interface. Delete the *Allow all ipv4+ipv6 via pfSsh.php* rule, and apply those changes to the firewall policy. Pray that you didn't just lock yourself out. If you did, check out the sidebar, *Have you looked in the mailbox?* below.

Have you looked in the mailbox?

If you're here, it means one of a few things probably happened:

1. The IP address of the pfSense WAN interface changed because a static IP address or DHCP mapping wasn't applied
2. The IP address of the management workstation changed because a static IP or DHCP mapping wasn't applied
3. The *Better Anti-Lockout* Rule wasn't created correctly, or no longer applies because of 1 or 2 above (e.g., the management workstation, or pfSense WAN interface "moved" and the firewall rule no longer works)
4. One of the above happened, there is no VM snapshot, and you don't want to have to re-create your pfSense VM

pfSense has documentation on various ways to regain access to the webConfigurator, but due to our *Explicit Deny Any WAN* rule, and my insistence on accessing the lab and webConfigurator through the *WAN* interface of the firewall, most of them won't work in this situation. If you decided to NOT create the *Explicit Deny Any WAN* rule, things are much easier. You can just re-run `pfSsh.php playback enableallowallwan` command from the command-line of the pfSense VM's virtual console, then fix the firewall rule policy once you log back in to the webConfigurator. The rest of you? It's gonna be a little more complicated than that. Listen up:

A little-known fact about pfSense is that the firewall rules are physically stored on a file called `/tmp/rules.debug`. We're going to run a series of commands to create a new anti-lockout rule, strip out our *Explicit Deny Any WAN* rule temporarily, reload the firewall policy, then exit the shell. Open up a console session to the pfSense VM, and select option 8 to get a shell on the firewall. Run the following commands in this exact order:

```
easyrule pass wan tcp [management workstation IP] [pfSense WAN interface IP] 443
egrep -v "Explicit Deny Any WAN" /tmp/rules.debug > /tmp/rules.debug.1
pfctl -f /tmp/rules.debug.1
exit
```

In the `easyrule` command above, replace `[management workstation IP]` with the IP address of your workstation, and `[pfSense WAN interface IP]` with the IP address of the pfSense VM's WAN interface. For example, the version of this rule I would use for my home network is:

```
easyrule pass wan tcp 10.0.0.3 10.0.0.206 443
```

```
[2.4.5-RELEASE][root@pfSense.localdomain]/root: easyrule pass wan tcp 10.0.0.3 1
0.0.0.206 443
Successfully added pass rule!
[2.4.5-RELEASE][root@pfSense.localdomain]/root: egrep -v "Explicit Deny Any WAN"
/tmp/rules.debug > /tmp/rules.debug.1
[2.4.5-RELEASE][root@pfSense.localdomain]/root: pfctl -f /tmp/rules.debug.1
[2.4.5-RELEASE][root@pfSense.localdomain]/root: exit
```

14-33: These commands are used to define a firewall rule that will allow us access to the webConfigurator, strip out the *Explicit Deny Any WAN* rule from our firewall policy temporarily, then tell pfSense to load the firewall rules from a file where that rule is removed. This is a very temporary work-around. We need to use this temporary access to fix the firewall rule policy for the WAN interface, save those changes, then reboot the firewall.

If everything was successful, you should be able to log back in to the webConfigurator. Navigate back to *Firewall > Rules*, and fix the firewall rule policy on the WAN interface. You may notice that the *Explicit Deny Any* rule is still listed on the firewall rule policy for the WAN interface. **This trick we're doing is intended to be very temporary.** Rebooting the firewall will make the *Explicit Deny Any WAN* rule work again.

The firewall rule created via the `easyrule` command will be labeled *Easy Rule: Passed from Firewall Log View*, and by default, it will always be the lowest firewall rule on the stack – this means it's below our *Explicit Deny Any WAN* rule. According to everything you know about pfSense processes firewall rules (top to bottom), we should never be allowed to access the webConfigurator, right?

The reason you're allowed to access the webConfigurator is, because technically we're not using the firewall rule policy being displayed. The webConfigurator is not aware of the firewall rule changes we made on the command line. The policy we loaded with `pfctl -f` does not have the *Explicit Deny Any WAN* rule defined.

We need to remove the old *Better Anti-Lockout Rule*, and place this new firewall rule at the top of the rule stack. I also recommend editing the description of the rule generated by `easyrule`, and re-labeling it *Better Anti-Lockout Rule*. Once finished, click *Save* to apply the rule order changes, then *Apply Changes* to apply your new firewall policy.

Floating <u>WAN</u> LAN OPT1										
Rules (Drag to Change Order)										
■	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
■	✓ 0/0 B	IPv4 TCP	10.0.0.5	*	10.0.0.200	*	*	none		Better Anti-Lockout Rule
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	8000	*	none		Allow Splunk Web access to SIEM VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to IPS VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to Kali VM
■	✗ 0/0 B	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any WAN
■	✓ 1/138 KiB	IPv4 TCP	10.0.0.3	*	10.0.0.206	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View

14-34: If this were the firewall policy we were actually using, technically we shouldn't be allowed on the webConfigurator right now. Good thing we *technically* aren't using this firewall rule policy. Move the *Easy Rule: Passed from Firewall Log View* rule to the very top of firewall rule stack, delete the old *Better Anti-Lockout Rule*, and (optionally) rename the rule you created using the `easyrule` command in *fig. 14-33*.

Floating <u>WAN</u> LAN OPT1										
Rules (Drag to Change Order)										
■	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
■	✓ 1/206 KiB	IPv4 TCP	10.0.0.3	*	10.0.0.206	443 (HTTPS)	*	none		Better Anti-Lockout Rule
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.3	8000	*	none		Allow Splunk Web access to SIEM VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to IPS VM
■	✓ 0/0 B	IPv4 TCP	10.0.0.3	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to Kali VM
■	✗ 0/0 B	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any WAN

14-35: If you did everything in accordance to *fig. 14-34*, the end result should look something like this.

We're not quite done yet. In spite of us making the necessary changes to the WAN interface's firewall rule policy, we're still *technically* not using this firewall policy. The fastest way to fix that is to open of a console session to the pfSense VM, select option 5 to reboot the VM, then input `y` to confirm that you would like to reboot the pfSense VM.

```
Enter an option: 5

pfSense will reboot. This may take a few minutes, depending on your hardware.
Do you want to proceed?

Y/y: Reboot normally
R/r: Reroot (Stop processes, remount disks, re-run startup sequence)
S: Reboot into Single User Mode (requires console access!)
F: Reboot and run a filesystem check

Enter an option: y
```

14-36: We successfully modified the firewall policy for the WAN interface, but the temporary firewall policy we hacked together on the command line is still being used. Rebooting the firewall is the fastest way to make pfSense use the firewall policy we just finished editing.

Once the pfSense VM finishes rebooting, try logging into the webConfigurator using the WAN interface IP address, same as you normally would. If you're greeted with the pfSense login page, the firewall edits made were successful. If not, you may have some troubleshooting to do:

- Has the IP address of either the management workstation or WAN interface of the pfSense VM changed again?
- Did you enter the commands in *fig. 14-33* exactly as instructed? The `egrep` command in particular is case sensitive.
- Consider doing network troubleshooting (check network cables, verify local network connectivity, etc.)

14.5 Chapter Review

If students have made it this far, that means the pfSense VM for their lab environment on the hypervisor of their choice is (almost) fully configured. Students should have a pfSense VM serving DNS, NTP, and Squid HTTP proxy services, along with an ironclad firewall rule policy, customized for hosted and/or bare-metal hypervisor users. The only things left to do are to add static DHCP allocations for remaining virtual machines in the lab environment as they are created, then create a snapshot of the pfSense VM once it is fully configured to preserve this hard work. Students should return to the hypervisor setup guide they initially came from to finish creating the remaining lab virtual machines:

Client Hyper-V: [Section 9.5, p. 135](#)

VirtualBox: [Section 10.7, p. 249](#)

VMware Fusion: [Section 11.5, p. 352](#)

VMware Workstation: [Section 12.6, p. 457](#)

Vmware ESXi: [Section 13.7, p. 599](#)

Chapter 15 Patch Notes

-I've included a brief introduction to routing, why it matters, and how network routing interacts with other network functions and protocols. It's pretty simplistic, but the idea is to teach students that a lot of network functions and protocols are inter-related.

-In addition to a full introduction to static routing, students are provided with all of the commands required to set static routes in their environment on Windows, Linux and MacOS.

-Notified Linux and MacOS users that for reasons entirely beyond me, virtual interfaces for most hosted hypervisors do not persist between reboots. This means static routes can't be persisted either. Linked to the FlightCheck-Linux and FlightCheck-OSX shell scripts that can be used to reconfigure virtual network interfaces, then automatically rebuild static routes. This chapter features a link to github gists to both scripts, and direct copies of both scripts in this manuscript as well.

-Since Kali Linux is an exercise in frustration, it has the SSH service installed, but it is not enabled or started. I created a section that instructs students on how to address this immediately.

-Windows users are advised on how to configure custom themes on mRemoteNG. The vs2015dark theme is really nice for not burning out your eyeballs.

-Sometime between now and the first edition, ssh-keygen got ported over to Windows 10. In addition to generating SSH keys using puttygen, I've taken the liberty of teaching students how to take keys generated by ssh-keygen and convert them to .ppk format so they can be used with PuTTY and mRemoteNG.

-Students will be instructed to generate ED25519 keys now (as opposed to RSA) because it's the future, and copying/pasting 4096/8192-bit RSA public keys manually sounds like a lesson in pain to me.

-Added guidance on the different type of SSH host key mismatch/missing error messages students might encounter using mRemoteNG and/or WinSCP, what they mean, and that it's generally acceptable to ignore them for setting up their lab environment

-Added a note that for reasons entirely beyond me, the vi/vim implementation on Kali Linux doesn't follow the normal PuTTY terminal behavior in that right clicking in the terminal window will "paste" text stored on the Windows clipboard. Instructed students on how to work around that, if they are using the "vi" method for copying their public key to the Kali VM to enable key-based authentication.

-For students that are having a hard time getting key-based authentication working and don't really want to spend effort fixing it right now, I provided a list of recommendations to protect their credentials from being stolen from the mRemoteNG confCons.xml file – either store your

credentials in a password manager and use it the way it was intended to be used, or enable options to fully encrypt the connection file, and set a password for the connection file to minimize risk.

-For students configuring SSH access from Linux/macOS hosted hypervisors, we're switching from using the `alias` command to create connection strings, to utilizing the SSH `config` file, and setting up connection profiles with that instead. This is because SSH `config` Host designations and configurations can be applied to `scp` and `ssh-copy-id`, making it infinitely more useful (for the purposes of this chapter anyway. The `alias` command still rocks)

-For feature parity, there is a section in which students are still taught on how to create ssh connection strings using the `alias` command.

-For establishing key-based auth for Linux/macOS hosted hypervisors, students will get familiar with `ssh-copy-id`, a handy utility that automates all of the most finicky parts of copying SSH public keys to remote systems.

-Since `ssh-copy-id` is so awesome, the methods I'm teaching students to use for copying ssh keys from Linux/Unix hypervisor hosts to their lab VMs have changed. I'm instructing students to use `ssh-copy-id`, `scp`, or the copy/paste method.

-Included an entire dedicated section for troubleshooting key-based authentication difficulties for both Windows and Linux/macOS students.

- Named the final section (*Optional*) *Remote Access Enhancements*. This section contains information on how to enable SSH access as the `root` user, as well as an entire section on how to disable password authentication over SSH completely.

-Students are advised that remote access as the `root` user is a risk factor and something that is not encouraged in the real world.

-Students are also advised on the importance of backing up the `sshd_config` file, because once password auth over SSH is disabled, that means they'll only be able to log in via SSH using key-based auth. If the keys are lost... Well, I made sure to also provide instructions on how to restore the default `sshd_config` file as well.

-I chose to focus on using the `sed` command to edit the `sshd_config` file to accommodate students who are not comfortable using command-line text editors (e.g., `vi`, `nano`, `ed`, etc.), but highly advise students to at least try to learn the `vi` editor, since it's so ubiquitous

- As it turns out, disabling password authentication by setting `UsePAM` to `no` in `/etc/ssh/sshd_config` is a bad idea. Jeremi M. Gosney (@jmgosney), a former embedded Linux developer, and password cracking professional, provided some guidance on best practices for disabling password auth over SSH, incorporated into this chapter.

Chapter 15: Routing and Remote Access for Hosted Hypervisors

This chapter will teach students on how to establish SSH access to their lab virtual machines on hosted hypervisors. The remaining chapters (Chapter 16 for bare-metal hypervisors notwithstanding) are going to be operating on the assumption that students have SSH access to the SIEM, IPS, and Kali virtual machines. The chapter content will be divided into these general sections:

- Configuring Static Routes
- Enabling the SSH service on the Kali VM
- Remote Access for Windows Hypervisor Hosts
- Remote Access for MacOS/Linux Hypervisor Hosts
- Troubleshooting Remote Access
- Optional Features

Note: As a reminder, back in Chapter 1, [section 1.5](#) (pp. 22-26), students were provided with a list of software applications they should install on their Windows, Linux or MacOS host systems. **Please ensure the recommended software is installed.**

15.1 Routing Tables and Static Routes

When computers send network traffic, whether it's to a destination on a local network, or to a destination out on the internet, software called a routing table determines where that traffic is going to be sent in order to ultimately reach its destination.

In our lab environment, our hypervisor host is directly connected to the 172.16.1.0/24 network through the host-only virtual network adapter. The host's routing table "knows" that other IP addresses on the 172.16.1.0/24 network can be reached by sending the traffic out of the host-only virtual interface. This is why students running their lab environment on a hosted hypervisor were able to access the pfSense webConfigurator without having to worry about whether or not there was a route to reach it. However, from a networking perspective, the hypervisor host is entirely unaware of the 172.16.2.0/24 network, or that it can be reached through the pfSense firewall. This is where static routes, and the `route` command comes into play. Windows, Linux and MacOS all have some variation of the `route` command, but the syntax, or how the program works, varies on each operating system. Using the host operating system's `route` command, students will need to add a static route to the 172.16.2.0/24 network.

A static route is, more or less, a route manually added to the system's routing table. It tells the computer "For any traffic destined to this particular network or host, send the traffic to this IP address." This is in contrast to a default route (or default gateway), which tells the computer "For any destination IP address that is NOT in the routing table, send it to this destination and hopefully, it'll know where to forward the traffic to reach its destination."

By default, manually added static routes do not persist. That means if a system with a bunch of static routes defined is rebooted, all of those routes will be gone. However, there are methods available on most operating systems to overcome this problem. For example, the Windows route command has an option to persist routes by saving them to the Windows registry, then re-applying them on boot. Linux and MacOS have the option to automate the creation of static routes through scripting, and (optionally) applying these routes on boot—with some exceptions that, unfortunately, we'll need to discuss further. In the coming subsections, students will create static routes to the 172.16.2.0/24 network through their host operating system's route command.

Preparing Routes

Before we continue on, ensure the following tasks have been performed:

- **Ensure that the host-only virtual network adapter on the Windows, Linux, or MacOS hypervisor host is present**
- **Make sure the host-only virtual network adapter is fully configured with the correct IP address and subnet mask.** Usually, this should be IP address 172.16.1.2, and netmask 255.255.255.0 (/24)
- **The pfSense virtual machine MUST be powered on and should be fully configured for the static route to function correctly.** This means that you should have performed the initial setup tasks for pfSense in the hypervisor setup chapter(s) (e.g., installing the OS and providing the WAN, LAN and OPT1 interfaces with IP addresses), and ALL of the remaining setup tasks in Chapter 14 *before* attempting to create a static route

Refer back to the hypervisor setup chapter for your host operating system and hypervisor if you need a refresher on how to set the IP address of the host-only virtual network adapter:

Microsoft Client Hyper-V: Chapter 9, section 9.3.3, pp. 100-101

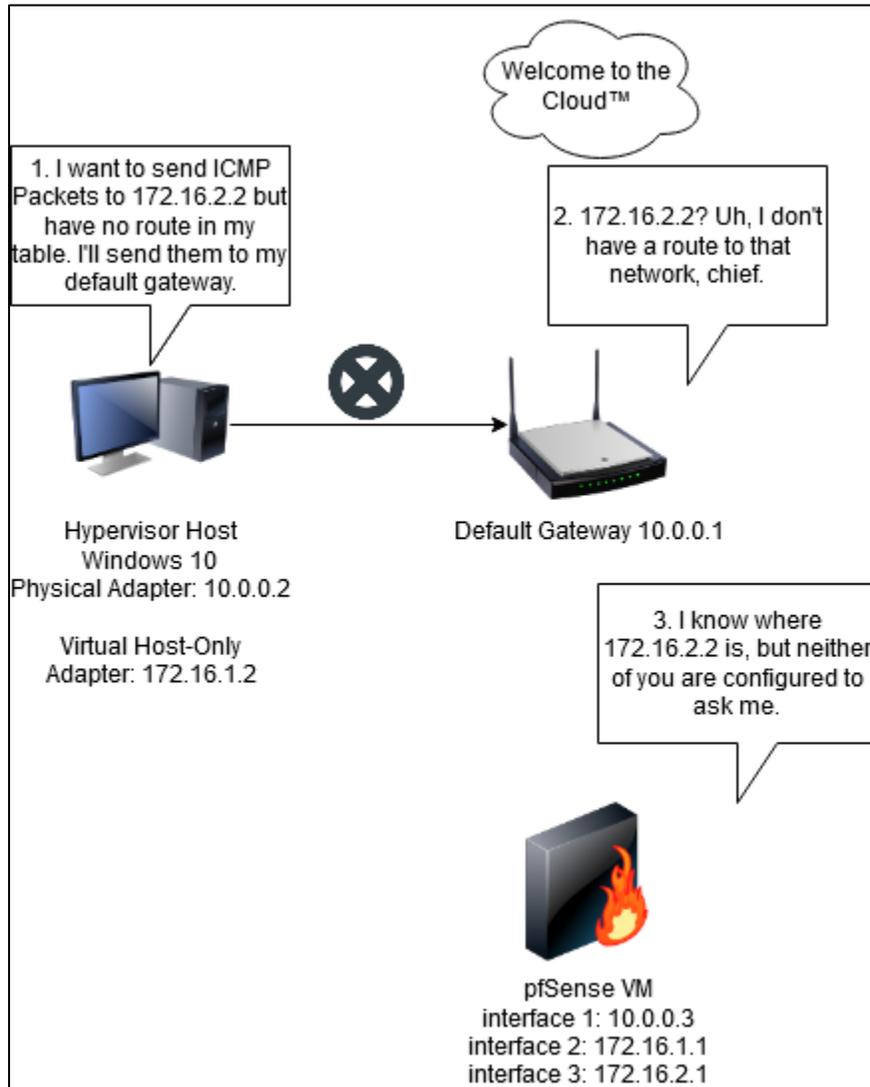
Oracle VirtualBox: Chapter 10, section 10.5, pp. 210-215

VMware Fusion: Chapter 11, section 11.3, p. 317

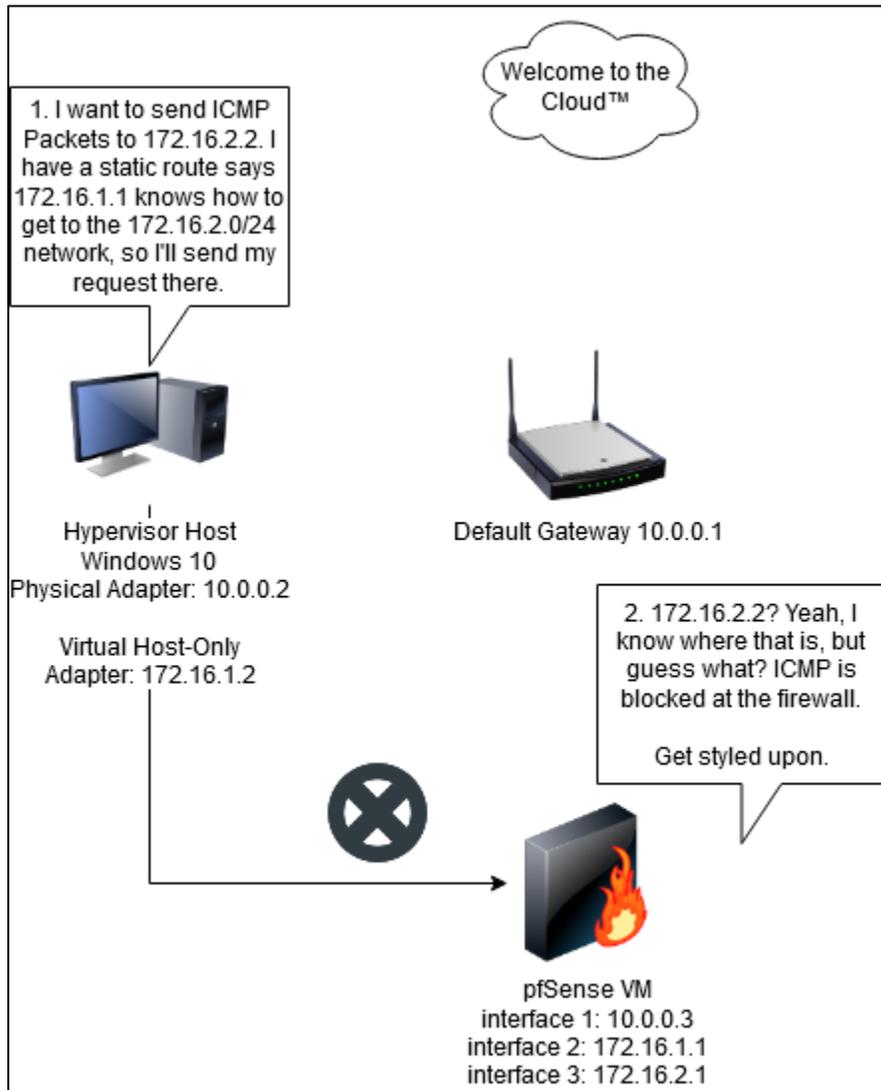
VMware Workstation: Chapter 12, section 12.4, pp. 423-426

And if you haven't finished setting up the pfSense virtual machine yet, jump to Chapter 14, starting on page 664.

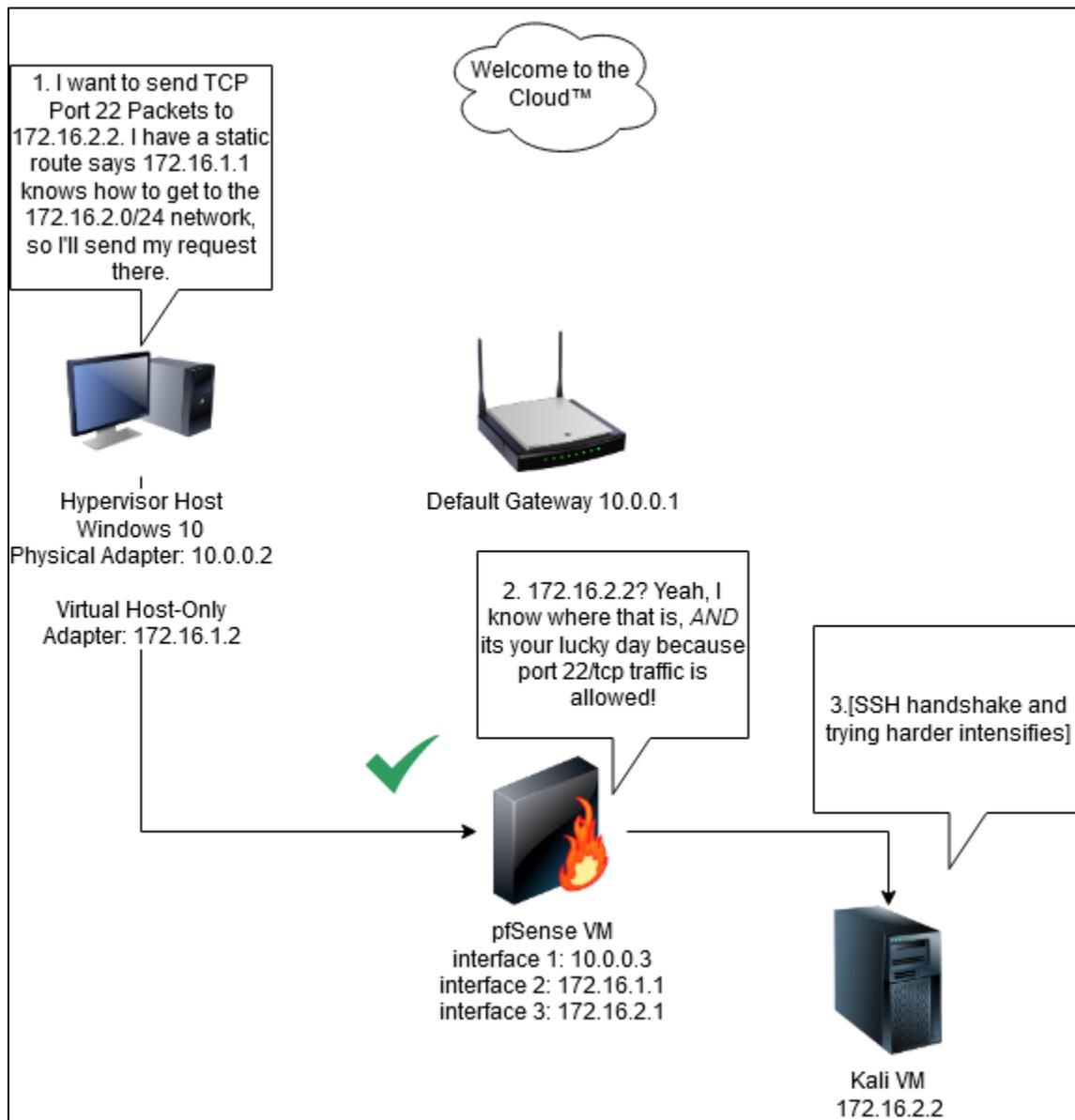
Also, as a general reminder, be aware that **host-only virtual network interfaces on Linux or MacOS have a tendency to disappear and/or lose their IP address and subnet mask configuration on reboot.** This means that if the machine has been rebooted recently, you'll need to start the hypervisor application (e.g., VirtualBox, or VMware Workstation/Fusion), reconfigure the IP address of the interface (e.g., vboxnet0, vmnet1 or vmnet2), AND power on the pfSense VM before attempting to create or use static routes.



15-1: While there is a lot more to it than this, network routes are a huge part of how computers figure out how to get data from point A to point B. The Windows host has no idea how to reach 172.16.2.2, so it asks its default gateway. The default gateway doesn't know either. The pfSense VM knows how to reach 172.16.2.2, and can be used to route traffic to that host, but the Windows host has no routes defined that tell it to ask the pfSense VM.



15-2: The purpose of this illustration is to demonstrate the point that ***routing is but a single component in how network traffic ultimately reaches its destination.*** Sure the Windows host has a route to 172.16.2.0/24, but there are no firewall rules in the pfSense firewall policy that allow ICMP, so ultimately the traffic *still* never makes it to the destination. Remember this when troubleshooting network problems in the future.



15-3: Again, this is very generalized, but in this diagram, the third try was the charm. The Windows host had a route to 172.16.2.0/24, port 22/tcp was allowed to host 172.16.2.2, and the host at that IP address had a service that was listening and willing to respond to the request. **Keep these diagrams in mind when attempting to troubleshoot network connectivity in the future** – is the destination reachable? Is there an on-path device (e.g. firewall, proxy, etc.) blocking the connection? Is the destination host up? Does the destination host have a firewall? Is it allowed to respond? Is there a service listening on the requested port? These are all important network troubleshooting questions.

15.1.1 Persistent Static Routes on Windows

Begin by opening the start menu. Click on the Windows search bar and type in 'cmd'. Right-click on the command prompt application, and choose *Run as Administrator*. If UAC is enabled, a UAC prompt will appear, asking if students want to run the command prompt with administrator rights. Click *Yes* to continue. In the command prompt, enter the command:

```
route -p add 172.16.2.0 mask 255.255.255.0 172.16.1.1
```

Then hit enter. If the command was entered successfully, students will see the output OK! Appear immediately below the command. This confirms that we have added a persistent route to the 172.16.2.0/24 network, via 172.16.1.1, the pfSense VM LAN interface.

Note: As a reminder, if students are utilizing alternate network address ranges for their lab environment, be sure to substitute 172.16.2.0 with the network address of the OPT1 (IPS) network, and 172.16.1.1 with the IP address of the LAN interface of the pfSense VM as necessary.

Next, run the command:

```
route print -4
```

If everything went well, towards the bottom of the window, there will be a section labeled *Persistent Routes*, and the route to the 172.16.2.0 network should be displayed. When finished, type *exit* to close the command prompt.

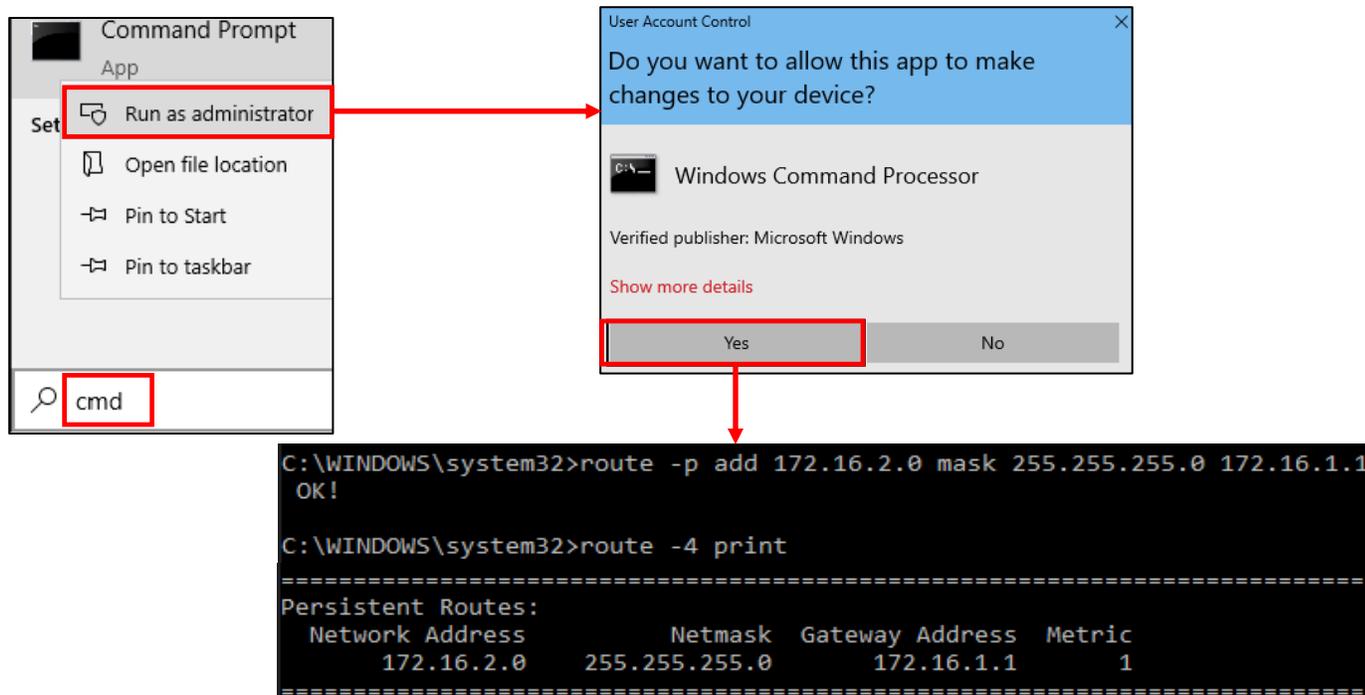
Fantastic Routes (and Where to Find Them)

If you're the type of reader who enjoys poking about in the Windows registry, and want to know where static routes are stored (when the *-p* option is used), they can be found at the following registry key:

```
HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRoutes
```

This is according to the official Microsoft documentation for the route command⁴.

⁴ https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/route_ws2008



15-4: to create a persistent static route, click *Start*, and type *cmd* in the search bar. Right-click on the *Command Prompt* app, and select *Run as administrator*. When the UAC prompt appears, Select *Yes*. Then run the commands:

```
route -p add 172.16.2.0 mask 255.255.255.0 172.16.1.1
route -4 print
```

If the output of those commands matches the output above (**note**: the `route -4 print` output above was *heavily* truncated), then adding the persistent static route was successful.

15.1.2 Static routes on Linux

Students should open their preferred terminal application, and run the following commands:

```
sudo ip route add 172.16.2.0/24 via 172.16.1.1
ip -4 route
```

The first command uses `sudo` to add a route to the 172.16.2.0/24 network, through the pfSense VM's LAN interface at 172.16.1.1. The second command is to confirm that the route was successfully added to the routing table.

Note: As a reminder, if students are utilizing alternate network address ranges for their lab environment, be sure to substitute 172.16.2.0 with the network address of the OPT1 (IPS) network, and 172.16.1.1 with the IP address of the LAN interface of the pfSense VM as necessary.

```

ayy@ayy:~$ sudo ip route add 172.16.2.0/24 via 172.16.1.1
[sudo] password for ayy:
ayy@ayy:~$ ip -4 route
default via 10.0.0.1 dev ens160 proto dhcp metric 100
10.0.0.0/24 dev ens160 proto kernel scope link src 10.0.0.85 metric 100
169.254.0.0/16 dev ens160 scope link metric 1000
172.16.1.0/24 dev vboxnet0 proto kernel scope link src 172.16.1.2
172.16.2.0/24 via 172.16.1.1 dev vboxnet0

```

15-5: Open a terminal window and run the commands:

```

sudo ip route add 172.16.2.0/24 via 172.16.1.1
ip -4 route

```

The output of the `ip -4 route` command can be used to confirm the route has been added to the routing table successfully. Look for the text `172.16.2.0/24 via 172.16.1.1 dev [host-only interface name]`.

15.1.3 Static Routes on MacOS

With both Windows and Linux covered, that leaves MacOS. Open either iTerm or iTerm2 and run the following commands:

```

sudo route add 172.16.2.0/24 172.16.1.1
netstat -nr -f inet | grep 172.16.2

```

The first command uses `sudo` to add a route to the `172.16.2.0/24` network, through the pfSense VM's LAN interface at `172.16.1.1`. The second command is to confirm that the route was successfully added to the routing table. If the route command was successful, a single line should pop up similar to the one in fig. 15-6 below.

Note: As a reminder, if students are utilizing alternate network address ranges for their lab environment, be sure to substitute `172.16.2.0` with the network address of the OPT1 (IPS) network, and `172.16.1.1` with the IP address of the LAN interface of the pfSense VM as necessary.

```

trobinson@trobinsons-MacBook-Pro ~ % sudo route add 172.16.2.0/24 172.16.1.1
Password:
add net 172.16.2.0: gateway 172.16.1.1
trobinson@trobinsons-MacBook-Pro ~ % netstat -nr -f inet | grep 172.16.2
172.16.2/24      172.16.1.1      UGSc      vmnet2

```

15-6: Open a terminal window and run the commands:

```

sudo route add 172.16.2.0/24 172.16.1.1
netstat -nr -f inet | grep 172.16.2

```

The output of the `netstat` command will look similar to what you see above. You'll notice that the last field reads `vmnet2`. I made this example using VMware Fusion. This field will read `vboxnet0` if ran with VirtualBox as the hypervisor.

15.1.3.1 *flightcheck-Linux and flightcheck-OSX*

Back in the hypervisor setup chapters (and in the sidebar conversation above, *Preparing Routes*), it was mentioned that most virtual interfaces from hosted hypervisors on Linux or MacOS, for one reason or another, fail to persist through reboots. For example in Chapter 10 (Virtualbox), `vboxnet0` is the host-only virtual network interface students configure on Linux and/or MacOS. The moment the system is rebooted, `vboxnet0` disappears until VirtualBox is started. Even then, the network configuration for the interface is gone: IP addresses, subnet masks, and static routes need to be re-applied every single time the system is restarted.

While I still don't have a solution to this problem, I do have some scripts I wrote that will automate a couple of tedious tasks:

- Reconfigures the IP address and subnet mask of `vboxnet0` (VirtualBox on Linux and MacOS), `vmnet1` (VMware Workstation on Linux), or `vmnet2` (VMware Fusion on MacOS)
- Creates a static route to the `172.16.2.0/24` network via `172.16.1.1`

I call these scripts FlightCheck, and there are two versions available: `flightcheck-Linux` and `flightcheck-OSX`, for Linux and MacOS, respectively. There are two github gists hosting the body of the scripts:

`flightcheck-Linux`: <https://gist.github.com/da667/ce39d245bd9d6bd30a205156a48ec67e>

`flightcheck-OSX`: <https://gist.github.com/da667/1a71ac4e85867a75785291c53306237f>

Because the internet is a mish-mash of technologies and companies who can up and decide that they no longer wish to play, removing access to useful tools and resources (like the links above), I'm going to make a copy of both of these simple shell scripts here in this book (with a bunch of the comment lines stripped out), so you can manually type it out, if there is a need to, or students can try to copy and paste from a PDF, if you have a digital version of this book.

If students decide to copy the script from the PDF, be aware of weird formatting glitches that can occur because PDFs are an abomination against nature. Last but not least, **make absolutely sure your text editor saves the file with Unix line feeds (LF) and not Windows line feeds (CR LF)**. As an example, both BBEdit (MacOS) and Notepad++ (Windows) have ways to confirm this setting.

What if I'm not using the default lab network ranges?

The flightcheck scripts assume students are using the networks 172.16.1.0/24 for the LAN/Management network, and 172.16.2.0/24 for the IPS/OPT1 network. If you are in a situation where you are using an alternative network range to avoid IP address conflicts, you'll need to make a couple of changes:

- Linux users will need to change the `ip addr add` command for either `vmnet1` (VMware Workstation) or `vboxnet0` (Oracle Virtualbox).
- Linux users will also need to modify the `ip route add` command to reflect the new IP address of the pfSense LAN interface, as well as the network address of the IPS/OPT1 network.
- MacOS users will need to modify the `ifconfig` command for `vmnet2` (VMware Fusion) or `vboxnet0` (Oracle VirtualBox)
- MacOS users will also need to modify the `route add` command to reflect the IP address of the pfSense LAN interface, and network address of the IPS/OPT1 network.

For example, let's say you're running VMware Fusion on MacOS. And you are using the network range 10.0.1.0/24 for the LAN/Management network, and 10.0.2.0/24 for the IPS/OPT1 network. The `vmnet2` host-only network adapter should be assigned the IP address 10.0.1.2, and the pfSense LAN interface should be assigned the IP address 10.0.1.1. So you should change the command:

```
ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
```

to:

```
ifconfig vmnet2 10.0.1.2 netmask 255.255.255.0
```

Then, change the command:

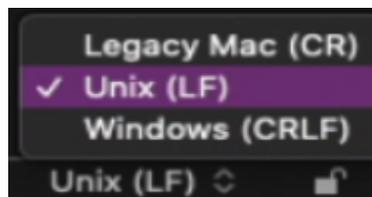
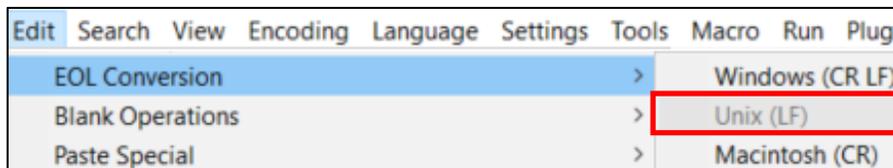
```
route add 172.16.2.0/24 172.16.1.1
```

to:

```
route add 10.0.2.0/24 10.0.1.1
```

```
ayy@ayy:~/Downloads$ sudo bash flightcheck-Linux.sh
[sudo] password for ayy:
Warning: Please make sure that all of your vmware workstation OR oracle virtualbox
VMs are running. In particular, the pfSense VM MUST be running!
Once you have verified that your VMs are up and running, Press enter to continue.
Checking for root privs..
We are root.
Checking to see if vmnet1 exists...
vmnet1 interface does not exist. Checking to see if vboxnet0 exists...
vboxnet0 interface exists. Flushing current IPv4 address...
Setting IP to 172.16.1.2/24...
vboxnet0 interface IP set to 172.16.1.2
Adding static route to 172.16.2.0/24 via 172.16.1.1...
added route to 172.16.2.0/24 via 172.16.1.1.
```

15-7: I wrote a pair of scripts for Linux and MacOS users called flightcheck-Linux and flightcheck-OSX. In a nutshell, these scripts will look the host-only network interface, configure it with the IP address 172.16.1.2/24, then add a route to the system's routing table to the 172.16.2.0/24 via the LAN interface of the pfSense VM (172.16.1.1).



15-8: For students who plan on copying the flight check shell scripts out of this book manually or via copy/paste, **make absolutely sure that the EOL (End of Line) conversion feature for your text editor is set to Unix (LF)**. To do this on Windows in Notepad++, click *Edit* from the navigation menu, select *EOL Conversion*, and ensure the *Unix (LF)* option is elected. For BBEdit users on MacOS, there is a small drop-down menu at the bottom of the text editor window that allows users to select the EOL conversion characters they want to use for the current document. Ensure that the *Unix (LF)* option is select. For students running text editors on Linux, this shouldn't be a problem you need to worry about.

Flightcheck-OSX.sh

```
#!/bin/bash
echo "Warning: Please make sure that all of your VMware Fusion OR Oracle VirtualBox
VMs are running. In particular, the pfSense VM MUST be running!"
read -p "Once you have verified that your VMs are up and running, Press enter to
continue."

echo "Checking for root privs.."
if [ $(whoami) != "root" ]; then
    echo "This script must be ran with sudo or root privileges."
    exit 1
else
    echo "We are root."
fi

echo "Checking to see if vmnet2 exists..."
ifconfig vmnet2 > /dev/null 2>&1
if [ $? -eq 0 ]; then
    echo "vmnet2 interface exists. Setting IP to 172.16.1.2/24..."
    ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
    if [ $? -eq 0 ]; then
        echo "vmnet2 interface IP set to 172.16.1.2."
    else
        echo "Could not set IP address of vmnet2 interface. You got troubleshooting
to do. Aborting."
        exit 1
    fi
else
    echo "vmnet2 interface does not exist. Checking to see if vboxnet0 exists..."
    ifconfig vboxnet0 > /dev/null 2>&1
    if [ $? -eq 0 ]; then
        echo "vboxnet0 interface exists. Setting IP to 172.16.1.2/24..."
        ifconfig vboxnet0 172.16.1.2 netmask 255.255.255.0
        if [ $? -eq 0 ]; then
            echo "vboxnet0 interface IP set to 172.16.1.2."
        else
            echo "Could not set IP address of vboxnet0 interface. You got
troubleshooting to do. Aborting."
            exit 1
        fi
    else
        echo "vboxnet0 interface does not exist. This script only checks for vmnet2
or vboxnet0. Aborting."
        exit 1
    fi
fi
fi
```

```
echo "Adding static route to 172.16.2.0/24 via 172.16.1.1..."
route add 172.16.2.0/24 172.16.1.1
if [ $? -ne 0 ]; then
    echo "Could not create route to 172.16.2.0/24. Does the network exist? Is vmware
fusion running? Is virtualbox running? Is the pfSense/gateway VM running? Does the
pfSense LAN interface have the correct IP address?"
    exit 1
else
    echo "added route to 172.16.2.0/24 via 172.16.1.1."
fi
exit 0
```

Flightcheck-Linux.sh

```
#!/bin/bash
echo "Warning: Please make sure that all of your vmware workstation OR oracle
virtualbox VMs are running. In particular, the pfSense VM MUST be running!"
read -p "Once you have verified that your VMs are up and running, Press enter to
continue."

echo "Checking for root privs.."
if [ $(whoami) != "root" ]; then
    echo "This script must be ran with sudo or root privileges."
    exit 1
else
    echo "We are root."
fi

echo "Checking to see if vmnet1 exists..."
ip addr show dev vmnet1 > /dev/null 2>&1
if [ $? -eq 0 ]; then
    echo "vmnet1 exists. Flushing current IPv4 address..."
    ip -4 addr flush label "vmnet1" > /dev/null 2>&1
    echo "Setting IP to 172.16.1.2/24..."
    ip addr add 172.16.1.2/24 dev vmnet1
    if [ $? -eq 0 ]; then
        echo "vmnet1 interface IP set to 172.16.1.2"
    else
        echo "Could not set IP address of vmnet1 interface. You got troubleshooting
to do. Aborting."
        exit 1
    fi
else
    echo "vmnet1 interface does not exist. Checking to see if vboxnet0 exists..."
    ip addr show dev vboxnet0 > /dev/null 2>&1
    if [ $? -eq 0 ]; then
        echo "vboxnet0 interface exists. Flushing current IPv4 address..."
        ip -4 addr flush label "vboxnet0" > /dev/null 2>&1
        echo "Setting IP to 172.16.1.2/24..."
        ip addr add 172.16.1.2/24 dev vboxnet0
        if [ $? -eq 0 ]; then
            echo "vboxnet0 interface IP set to 172.16.1.2"
        else
            echo "Could not set IP address of vboxnet0 interface. You got
troubleshooting to do. Aborting."
            exit 1
        fi
    else
        echo "vboxnet0 interface does not exist. This script only checks for vmnet1
or vboxnet0. Aborting."

        exit 1
    fi
fi
```

```
echo "Adding static route to 172.16.2.0/24 via 172.16.1.1..."
ip route add 172.16.2.0/24 via 172.16.1.1

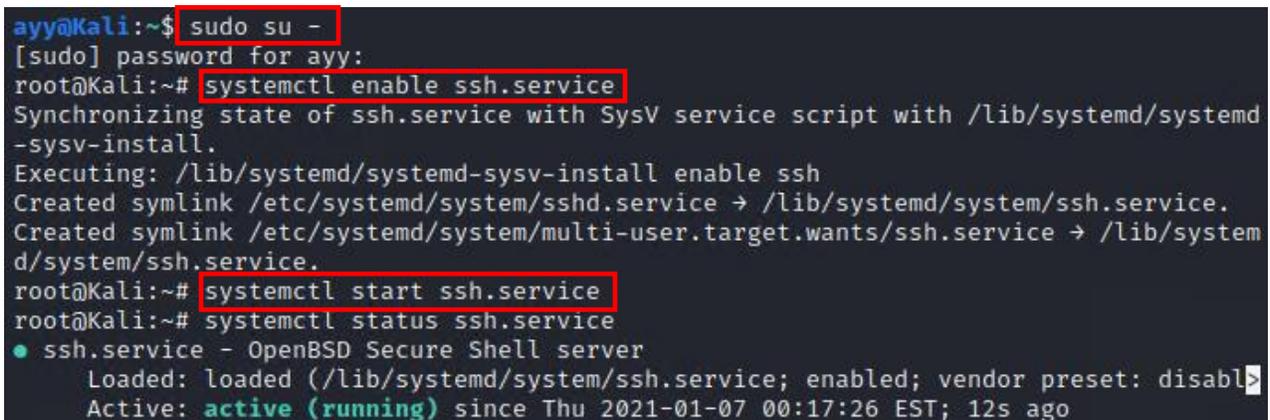
if [ $? -ne 0 ]; then
    echo "Could not create route to 172.16.2.0/24. Does the network exist? Is vmware
fusion running? Is virtualbox running? Is the pfSense/gateway VM running? Does the
pfSense LAN interface have the correct IP address?"
    exit 1
else
    echo "added route to 172.16.2.0/24 via 172.16.1.1."
fi
exit 0
```

15.1.4 Enabling SSH access on Kali Linux

In order to enable SSH access on the Kali Linux VM, students will need to enable the SSH service. This is because in spite of having the SSH service installed by default in Kali Linux, the service is not running, and it is disabled from starting up automatically on system boot, unlike the Ubuntu Server virtual machines, SIEM and IPS. Open a virtual console session to the Kali VM and log in, using the username and password students assigned when they installed the operating system. Once logged in, open a terminal session, and run the following commands:

```
sudo su -
systemctl enable ssh.service
systemctl start ssh.service
```

When finished, students may exit the terminal session and virtual console. I would highly recommend creating a new virtual machine snapshot or checkpoint for the Kali VM after enabling SSH to serve as a new baseline configuration. So, what did these commands do, exactly? We became the root user, used `systemctl` to tell Kali to allow the ssh service to start when the system starts up, then immediately told `systemctl` to start the SSH service right now. Since this represents a significant configuration change for the kali VM, taking a snapshot/checkpoint to reflect this configuration change is highly recommended.



```
ayy@Kali:~$ sudo su -
[sudo] password for ayy:
root@Kali:~# systemctl enable ssh.service
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-
sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/system
d/system/ssh.service.
root@Kali:~# systemctl start ssh.service
root@Kali:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabl
   Active: active (running) since Thu 2021-01-07 00:17:26 EST; 12s ago
```

15-9: In order to be able to connect to Kali over SSH, students will need to open a console session to the Kali VM on their hypervisor, log in, open a terminal session then enter the commands specified above to re-enable then start the ssh service. **After running these commands, take a new baseline snapshot of the Kali VM.**

15.2 Remote Access for Windows Hypervisor Hosts

In this section, students will learn how to configure remote access to their lab virtual machines hosted on Windows, using mRemoteNG. Students will also learn how to generate SSH keys, and apply them to their mRemoteNG connection profiles for key-based authentication. This will allow students to choose between double clicking on a connection profile and instantly getting a session on their VMs, or configuring two-factor authentication using a password-protected SSH key.

15.2.1 mRemoteNG

mRemoteNG is a Windows application that can be best described as a remote management aggregator. The main draw of mRemoteNG is that it's capable of handling a variety of remote access protocols such as VNC, RDP, SSH, Telnet, HTTP/HTTPS, etc. There is even limited support for telling mRemoteNG to launch specific external applications in order to handle proprietary remote connection protocols. Imagine having a single application that can be used to handle connecting to your Linux/BSD VMs via SSH, then by clicking a tab, you can switch to RDP sessions to your Windows VMs. What's more is that it's free software.

mRemoteNG handles connections to various systems and their various remote access protocols through the use of connection profiles. Connection profiles store the IP address or hostname of the remote system, username and/or password, and the protocol the connection should use. Once all of this information is saved to a profile, all that remains is to double click on the completed profile and if you provided the correct configuration (hostname, protocol, credentials) and solid network connectivity, a session will pop up instantly.

Customizing The Interface

Before we get into things, I'm going to make a couple of recommendations. These changes are entirely optional, but in order to de-clutter the interface a little bit, and not have your eyes burned out by all of the blinding whitespace on your screen, try some of these suggestions:

- In the navigation menu, select *View*, and in the drop-down menu that appears, disable any of the enabled toolbars. For example, on my system, the *Quick Connect Toolbar* was enabled. Clicking on the option if it's highlighted will remove it.
- In the navigation menu, select *Tools*, then *Options* in the drop-down menu. A new window appears labeled, *mRemoteNG Options*. Select the *Theme* option in the bottom of the window, ensure that the checkbox labeled *Enable Themes* is enabled, then, in the drop-down menu along the top of the window, locate and enable any of the many dark themes. Personally, I prefer the theme *vs2015dark*. Once you have made your choice, click *OK* to close the options menu, then restart mRemoteNG for the theme to apply itself properly.
- The *Connections* and *Config* panels on the left-hand side of the main window can be re-sized to make them smaller, and make the main window bigger.

15.2.2 Creating Connection Profiles

In the main mRemoteNG window, there are three panes. One of them is labeled *Connections*, another is labeled *Config*, and the third pane is the main screen for when you connect to hosts. Click on the *Connections* pane to highlight it. Hover over the white square furthest to the left with a tiny globe in the middle of it, and the text *New Connection* should appear. Click on this con, and a new connection profile will appear under the section labeled *Connections* with the small globe icon beside it.

The new connection profile, named *New Connection* should be highlighted by default. If not, click on it to highlight it, and a the *Config* pane should update with a huge variety of input boxes and configuration options. Change the following fields:

Name: SIEM

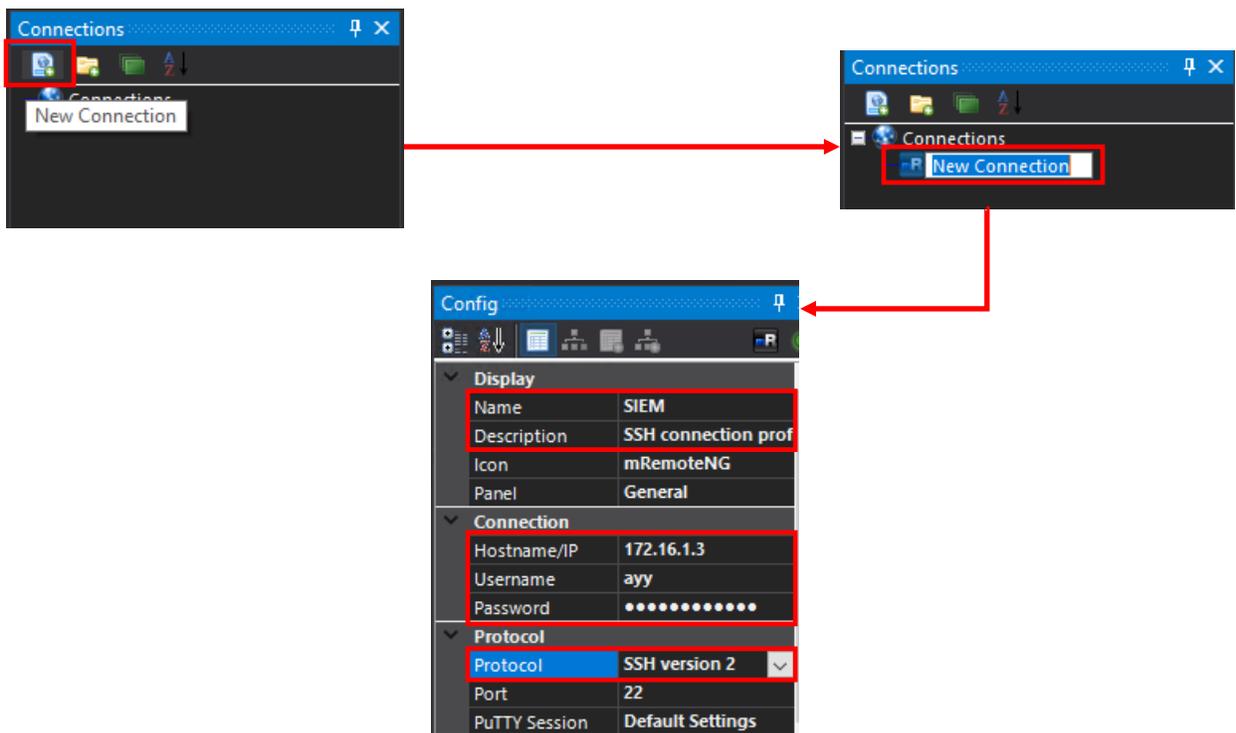
Description: SSH connection profile for SIEM VM

Hostname/IP: 172.16.1.3

Username: [username created during Ubuntu installation]

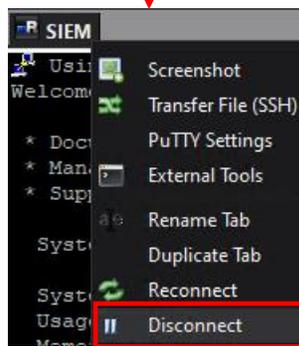
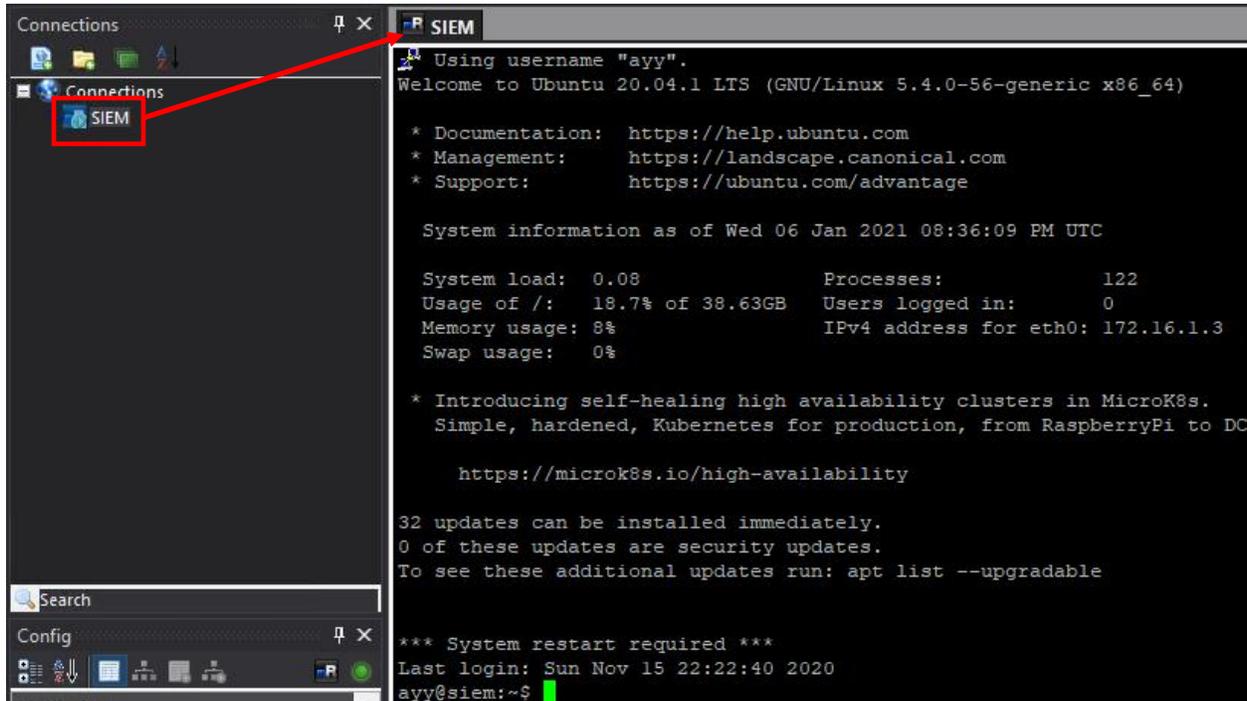
Password: [password created during Ubuntu installation]

Protocol: SSH version 2



15-11: Click the new connection profile icon, then click on the connect profile that appears, to highlight it. Fill out the *Name*, *Description*, *Hostname/IP*, *Username*, *Password* and *Protocol* fields as instructed above.

When finished, double click on connection profile labeled SIEM in the *Connections* pane. If students entered a valid username and password for the SIEM VM, they should be greeted with a welcome banner from the SIEM VM. When finished, type `exit` at the terminal prompt, or right-click on the tab labeled SIEM and select *Disconnect* to close the SSH connection to SIEM.

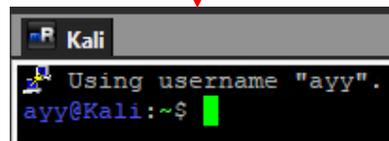
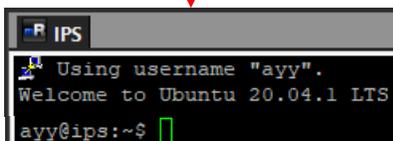
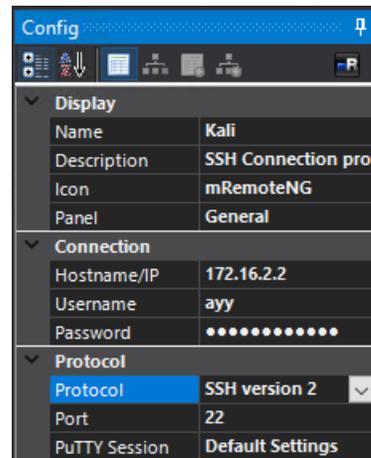
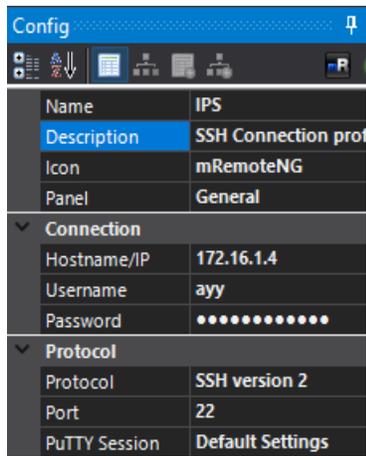


15-12: Double-click on the newly configured SIEM connection profile, and provided the credentials are valid, students should be greeted with an SSH login banner on the SIEM virtual machine. Students may use the `exit` command, or right click on the tab labeled *SIEM*, and select the *Disconnect* option to close the session.

Create two more connection profiles with the following information:

Name:	IPS	Kali
Description:	SSH Connection profile for IPS VM	SSH Connection profile for Kali VM
Hostname/IP:	172.16.1.4	172.16.2.2
Username:	[username created during Ubuntu installation]	[username created during Kali installation]
Password:	[password created during Ubuntu installation]	[password created during Kali installation]
Protocol:	SSH version 2	SSH version 2

Once finished, test the IPS and Kali connection profiles. SSH connections to the IPS virtual machine should work perfectly fine. If you are having problems connecting to the Kali Linux VM, please refer to [section 15.1.4](#), p. 742 for guidance on enabling and starting the SSH service. ***The SSH service must be enabled and started in order to accept SSH connections from mRemoteNG.*** Additionally, make absolutely sure the hypervisor host's host-only interface is configured with a valid IP address, subnet mask, and static route that allows access to the OPT1 network, in order to reach the kali VM. Configuring persistent static routes in Windows was covered in [section 15.1.1](#) (pp. 732-733).



15-13: Create additional connection profiles for the IPS and Kali VMs, then feel free to test them out. If students are experiencing problems connecting to the Kali VM, confirm that the steps specified in section 15.1.4 (p. 742) have been completed. The SSH service must be enabled and started in order to accept SSH connections.

SSH Host Keys and You

Some of you might have noticed that when you connect to your virtual machines that you might get security warnings similar to what is displayed below in *fig. 15-14*.

One version of this message will read:

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

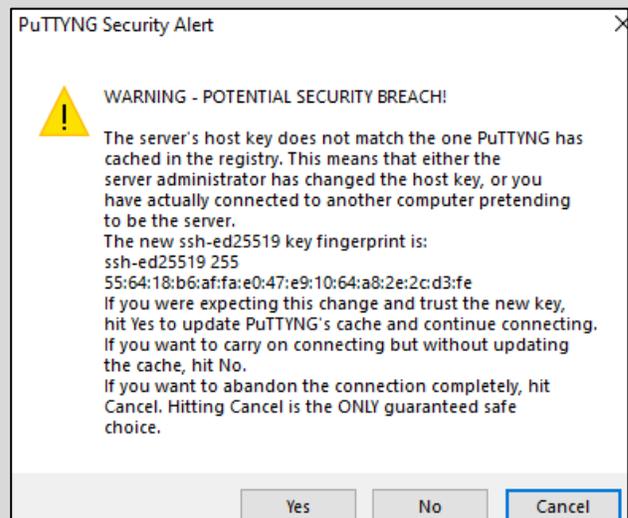
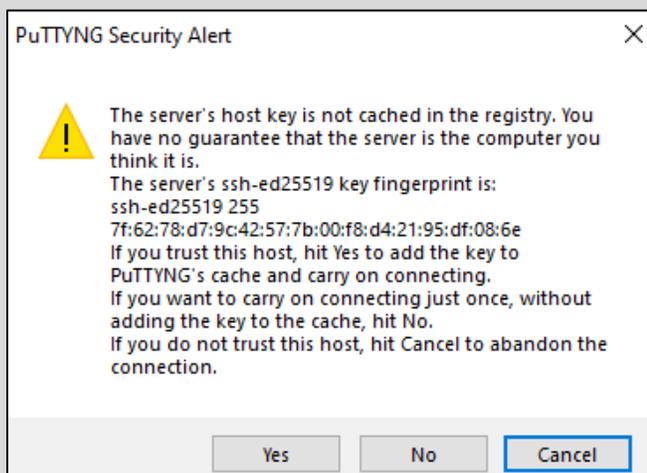
While another version of this message reads:

The server's host key does not match the one PuTTYNG has cached in the registry. This means that either the server administrator has changed the host key, or you have actually connected to another computer pretending to be the server.

The SSH protocol utilizes a feature called host key fingerprinting. Think of this as a way of positively identifying the server you are trying to connect to via SSH to help counter Man-in-the-Middle and/or Spoofing attacks.

The first version of this message warns users and tells them "Hey, there is an SSH server running on the IP address you want to connect to, but I don't have its host key on file. Are you sure you want to continue?"

The second message warns users, "There is an SSH server running on this host, but its host key fingerprint doesn't match the fingerprint I have on file." This can happen when you create a new virtual machine and it inherits the same IP address as a previous virtual machine whose SSH host key was already on file, so now the SSH client is convinced the new virtual machine is an imposter. In both instances, students should select *Yes* to continue connecting to their VMs.



15-14: Students will see one of these two messages when you first connect to their lab VMs over SSH. One error message is basically "This is a new host, do you wanna add it to your contacts?", while the other is "This host changed its phone number, and I think that's suspicious." Select *Yes* to continue, and Windows will either overwrite the SSH key fingerprint (if you created a new virtual machine) or add the new SSH key fingerprint to the registry as necessary.

15.2.3 Enabling Key-Based Authentication

The SSH protocol allows for a variety of different ways users can authenticate, or prove that they are the user they are trying to connect as. The most common authentication method is by pairing a username with a password (see section 15.2.2). In this section, we will be learning about key-based authentication for Windows hosts.

To make a long story short, SSH key-based authentication relies on two pieces of information to authenticate a user: A public key stored on the system you want to log in to over SSH, and a private key that your ssh client (e.g. mRemoteNG) must have access to. **The private key must be kept safe, and as the name implies, private. If someone else has your private key, they can log in as you.** Depending on how the private key is created, key-based authentication can be used on its own with no password to enable fast and easy remote access to your virtual machines, or the SSH private key can be password protected and require users to enter that password every time it is used to connect, providing a basic form of two-factor authentication (*something you have*, the SSH private key, and *something you know*, the password to decrypt and use that private key).

If students have not done so already, download and install both puttygen.exe, and Notepad++.

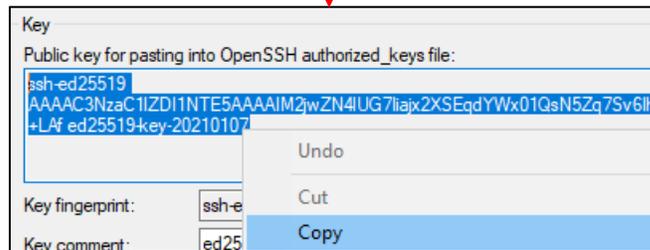
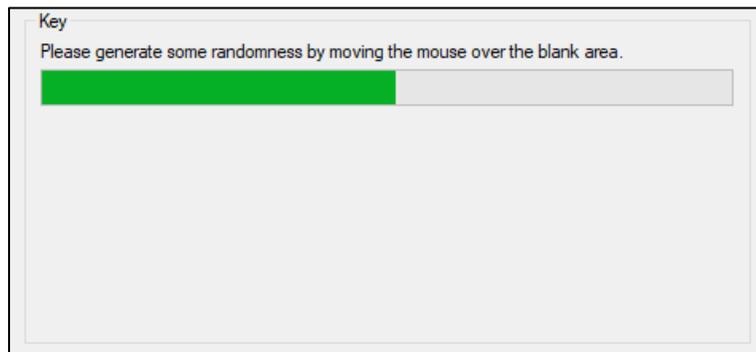
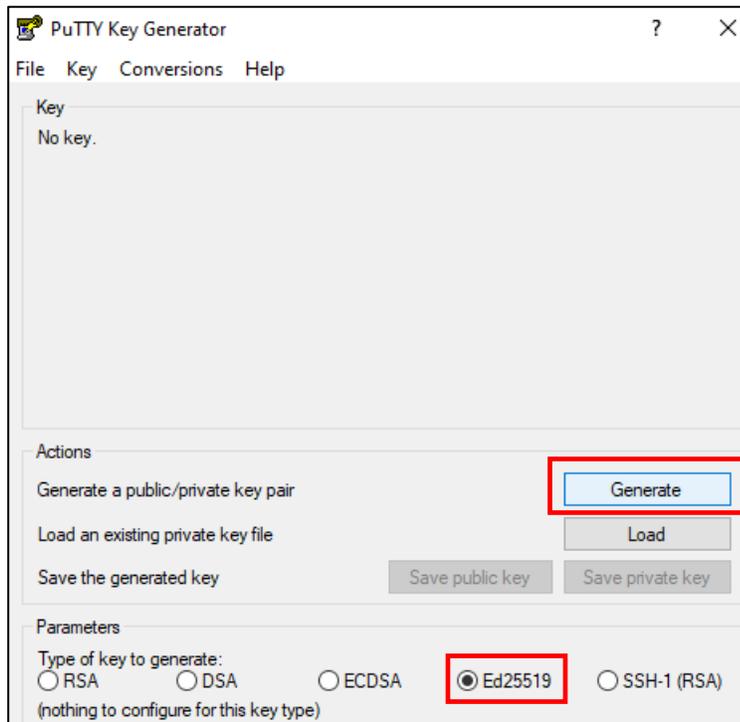
15.2.3.1 Generating Public and Private SSH keys using PuTTYgen

PuTTYgen is a simple executable that can be used to generate SSH keys on Windows. Double click on puttygen.exe and a new window, titled *PuTTY Key Generator*, will appear. In the *Parameters* section in the bottom of the window, click the radio button labeled *Ed25519*, then under the *Actions* section, click the *Generate* button. The *Key* section will update with a green bar and the text:

Please generate some randomness by moving the mouse under the blank area.

As a part of generating the SSH key, the puttygen application needs to generate the key using random numbers and entropy. The application relies on user input to generate enough entropy or "randomness" to create a secure key. After a moment or two, the *Key* section will update with a variety of input boxes and output.

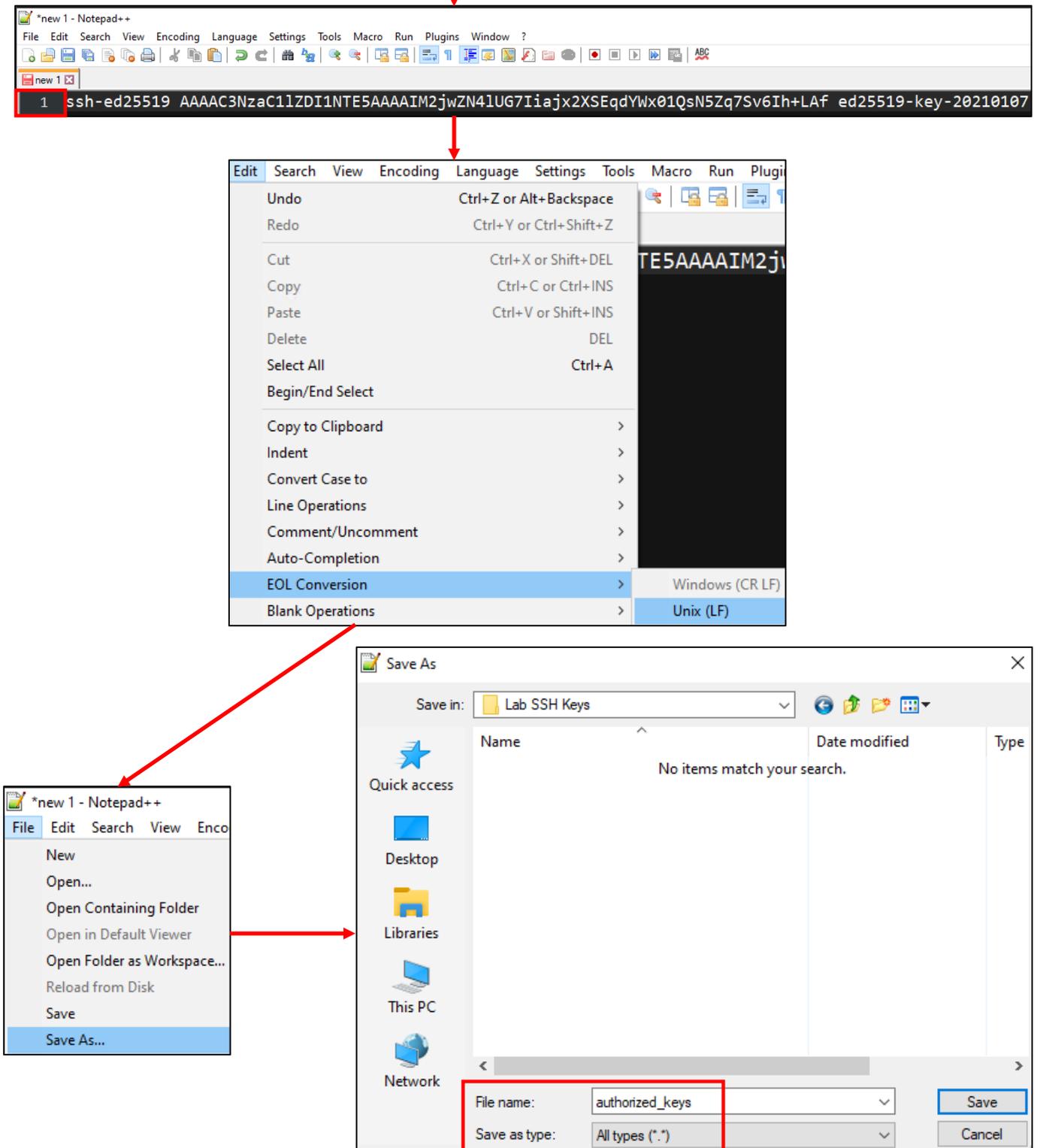
In the text box labeled *Public Key for pasting into OpenSSH authorized_keys file*, highlight and copy all of the text in the field, open Notepad++, then paste that text into a new file. **Make absolutely sure there is only one line in the notepad++ file.** Next, select *Edit* from the navigation menu, *EOL Conversion*, and *Unix (LF)*. **Since this is a file that students will be copying to Linux systems, the file must have the Unix (LF) end of line setting.** Next, select *File* from the navigation menu, followed by *Save As*. In the *Save As* window, save your edited file to an easy to access directory (e.g, I made a directory *Lab SSH Keys* on my desktop). In the *File name* field, name the file **authorized_keys**, and in the *Save as type* drop-down menu, select the option *All types (*.*)*, then click the *Save* button. Close Notepad++ and navigate back to the puttygen window.



Continued to *fig. 15-16*

15-15: Open puttygen.exe, Click on the *Ed25519* radio button under *Parameters*, then click on the *Generate* button under *Actions*. Move your mouse cursor under the *Key* section of the window to create entropy and help generate your SSH key. When finished, highlight and copy the entire section under the text *Public key for pasting into OpenSSH authorized_keys file*

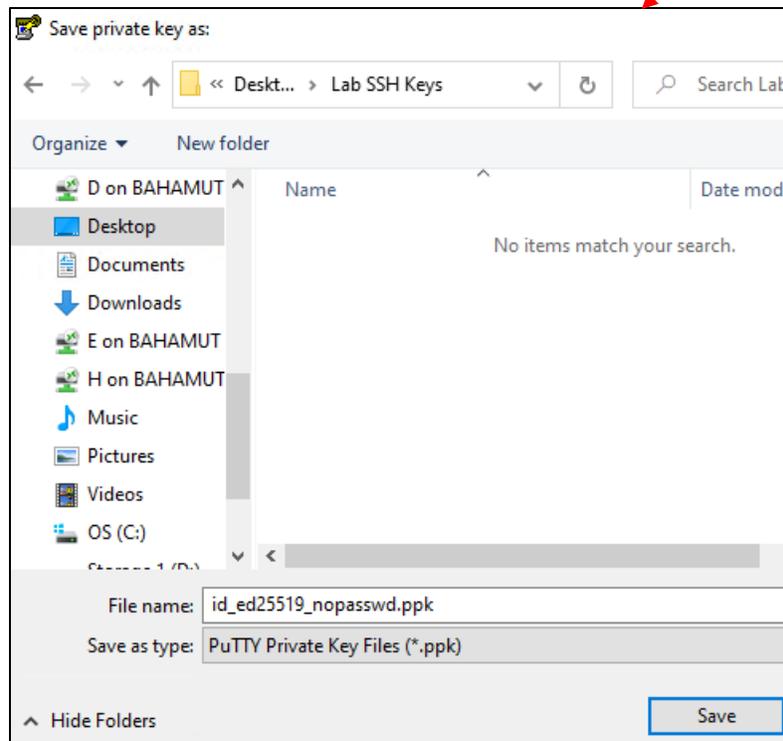
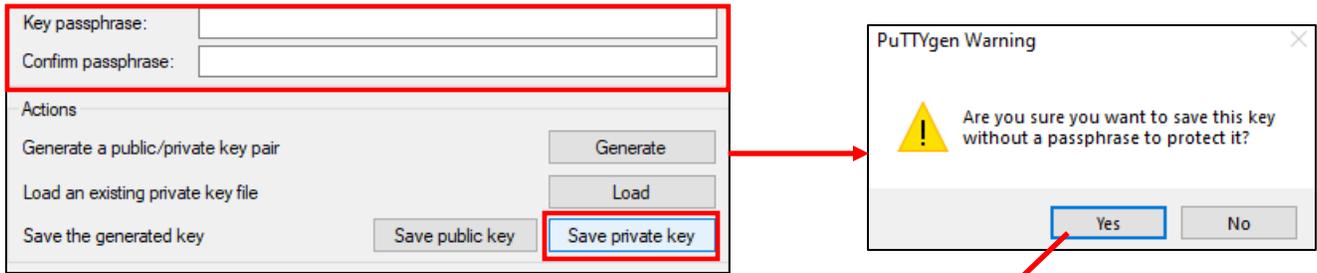
Continued from *fig. 15-15*



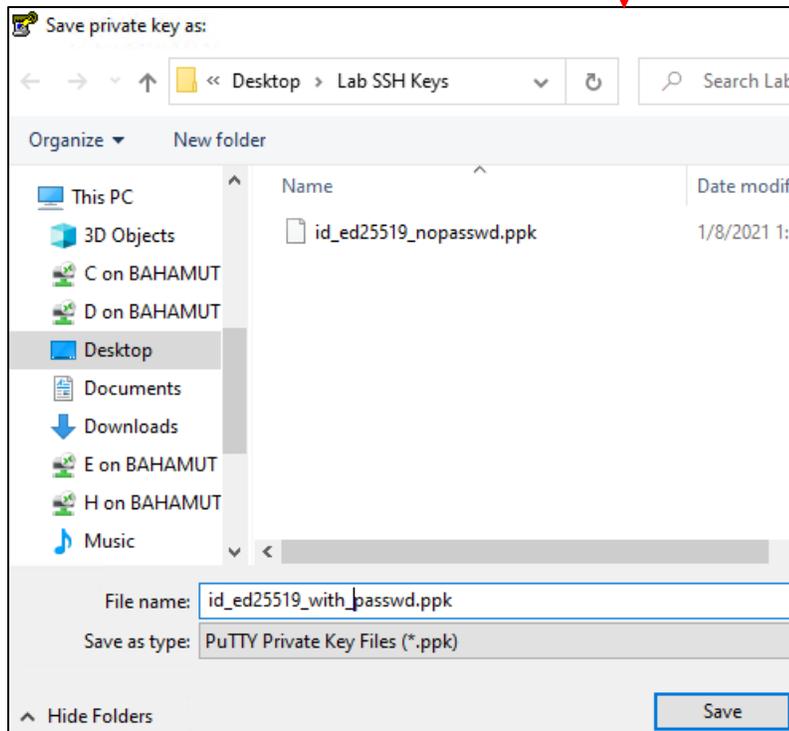
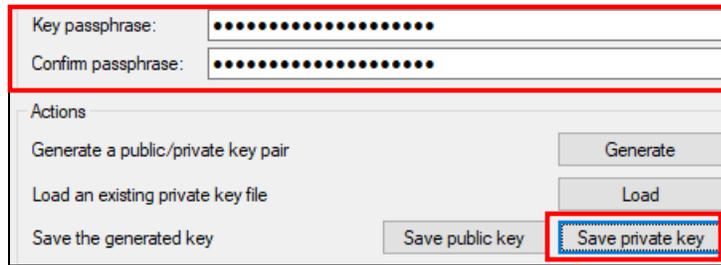
15-16: Next, paste the SSH public key copied from puttygen into Notepad++. Using the line numbers on the left margin of the Notepad++ window, ***make sure the file is only a single line long***. Afterwards, Select *Edit > EOL Conversion > Unix (LF)*. Finally, select *File > Save As*, and in the *File name* input box, enter `authorized_keys`, and in the *Save as type* drop-down menu, select *All types (*.*)*. Be sure to save this file to an easy to access location.

At the beginning of this section, I provided a brief introduction to key-based authentication and stated that depending on whether or not students password protect their private key, they could configure key-based authentication to be fast, convenient and passwordless, or harden SSH access to their virtual machines through two-factor authentication, requiring both access to the private key and the password to decrypt the private key for use. This is where the fields *Key passphrase* and *Confirm passphrase* under the *Key* section of the puttygen window come into play.

If students do not enter a password for their private key, and click the *Save private key* button under the *Actions* section of the window, puttygen will allow users to save a private key that is not password protected. This means that anyone with a copy of that private key can access any system configured to use key-based authentication with that private key. On the other hand, if students do enter a password in both of those fields, then click the *Save private key* button, both the private key, and the password to decrypt it are required to use key-based authentication on configured systems.



15-17: If students click the *Save private key* button under *Actions* with the *Key* and *Confirm passphrase* input boxes blank, this allows the creation of an SSH private key that is not password protected. puttygen will ask the user to confirm if this is what they want to do. If students click *Yes*, they are prompted to save their private key. The final image on this page is of an SSH session configured to use key-based authentication using a private key that is not password protected. mRemoteNG notifies the user that the the public/private key pair is being used to authenticate, and once successful, a session is instantly granted.



15-18: In comparison, if students elect to set a password on the private key, then click *Save private key*, and configure mRemoteNG to use the password-protected key, students will be prompted to enter that key's password when they log in. It's a very good idea to store the password to the private key in a password manager (e.g., KeePassXC).

Alternate Key Generation Method

Newer versions of Windows (Windows 10 and pre-release versions of Windows 11, at least) now ship with the `ssh-keygen` utility, a tool originally found on Unix/Linux-based operating systems. This command-line tool can be used to generate a public/private SSH key pair that users can choose to password protect or not, just like with `puttygen`. This sidebar conversation will teach you how to use the `ssh-keygen` command-line utility, then use `puttygen` to convert the private key created with `ssh-keygen` into a `.ppk` file for use with `mRemoteNG`.

First things first, create a folder in an easy to access location. For example, I created a folder on my desktop labeled, `ssh-keygen_keys`. The full path to this folder is: `D:\Tony\Desktop\ssh-keygen_keys\`. You'll need this full path information in a moment when you generate your SSH keys. This information can be obtained by double clicking on your newly created folder. A new explorer window will pop-up, and the title bar will display the full directory path.

Next, open a command prompt window by clicking *Start*, typing `cmd`, then clicking on the *Command Prompt* application. Run the `ssh-keygen` command with the following arguments:

```
ssh-keygen -t ed25519 -f [path to save SSH keys to]\[name of key file]
```

For example:

```
ssh-keygen -t ed25519 -f D:\Tony\Desktop\ssh-keygen_keys\lab_key_ed25519
```

This command will generate a public and private SSH key using the `ed25519` algorithm. The `ssh-keygen` command will ask if you want to assign a password to the private key. That choice is entirely up to you, as we discussed above. Password protecting your key makes your lab VMs a bit more secure by providing basic two-factor authentication, while not password protecting it will make SSH access more convenient. When finished, the `ssh-keygen` command will provide you with two files in the directory you specified:

```
[keyname]  
[keyname].pub
```

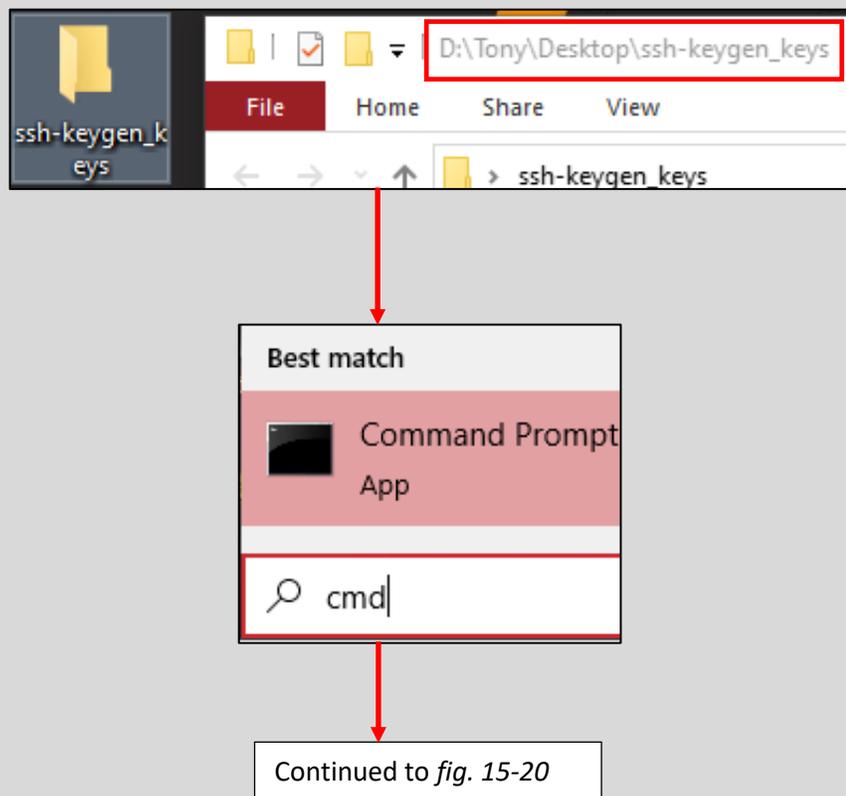
Which, from my example, are named:

```
lab_key_ed25519  
lab_key_ed25519.pub
```

Your next order of business involves opening the `.pub` file in `Notepad++` and making a couple of edits, namely, converting the EOL characters to Unix (LF), and saving the file as `authorized_keys`. If you installed *Notepad++* with all of its default options, you should be able to right click on the `.pub` file that was generated, and select *Edit with Notepad++* from the context menu. From there, take a look at the instructions provided in [section 15.2.3.1](#) (pp. 750-759), for a run-down on what needs to be done, and how to do it. Additionally, you can also check out [fig. 15-16](#) for further guidance.

With the public key properly formatted, the next, and final step is to convert the private key to a .ppk file to be used with mRemoteNG and PuTTYNG. Open the *puttygen* application and using the navigation menu at the top of the window, select *File > Load private key*. This opens an explorer window. You'll need to change the drop-down menu in the bottom-right corner from *PuTTY Private Key Files (*.ppk)* to *All files (*.*)* in order to be able to select your private key. Browse to the directory you used to store the generated keys from *ssh-keygen*, and select the [keyname] file. In my example, this would be the file, *1ab_key_ed25519*. Please be aware that if you decided to password protect your private key, that *puttygen* will prompt you with the password for that key in order to load it into *puttygen*.

With the private key loaded, click the *Save private key* button under *Actions*, and you'll be prompted with an explorer window to choose where to save the new .ppk file. Users who password protected their keys and correctly entered the password will have that password automatically populated into the Key passphrase and Confirm passphrase fields. If for some reason you've changed your mind and no longer wish to password protect your private key, delete the content contained in these fields.

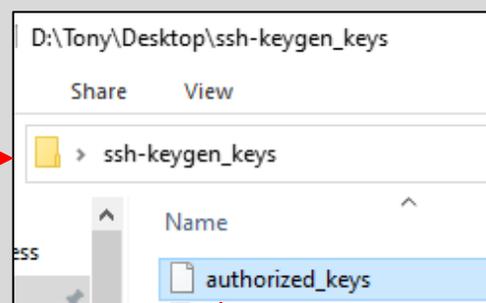
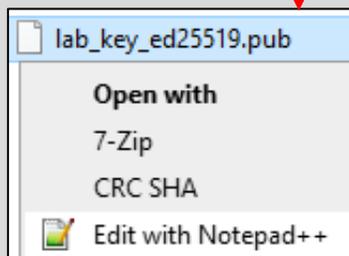
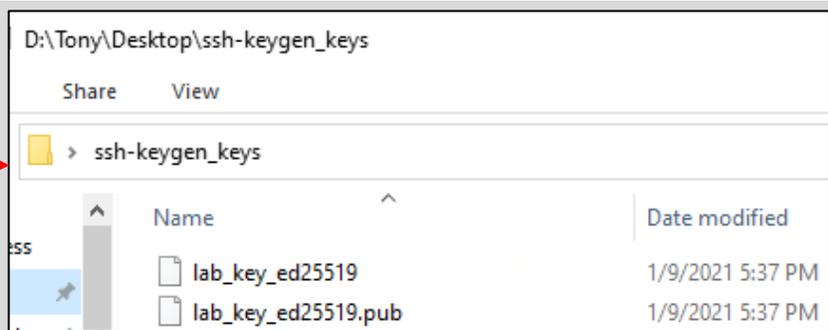


15-19: First things first, create a directory to store the keys that you'll be generating with *ssh-keygen*. Note the full directory path to the folder you created, because you'll need this in a moment. Next, open up a command prompt window. Usually, easiest way to do this is to use the windows search bar/start menu, type in *cmd*, and start the *Command Prompt* application, or if you're good at keyboard shortcuts: Windows/Meta key + R to open the run prompt, then type in *cmd.exe*.

Continued from *fig. 15-19*

```
Command Prompt
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

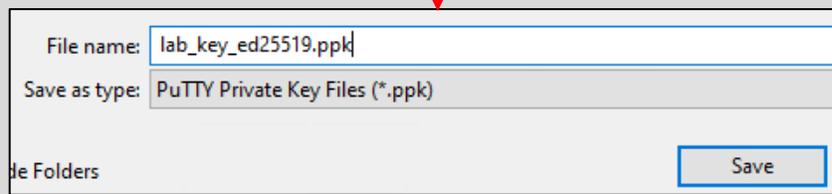
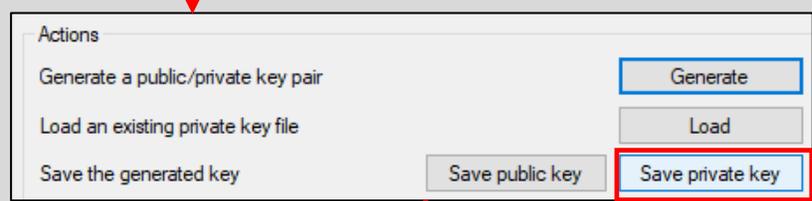
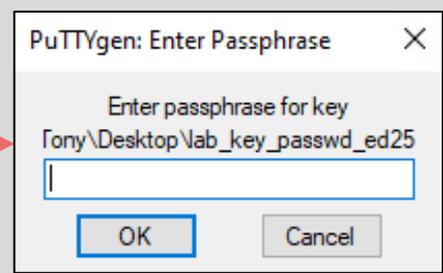
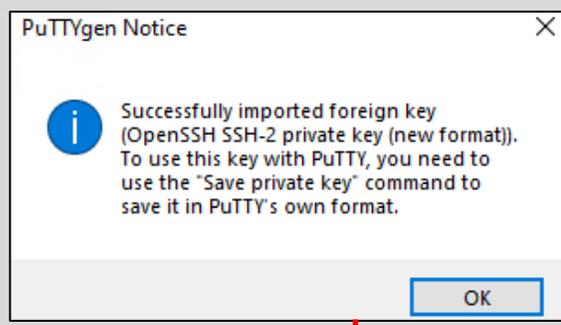
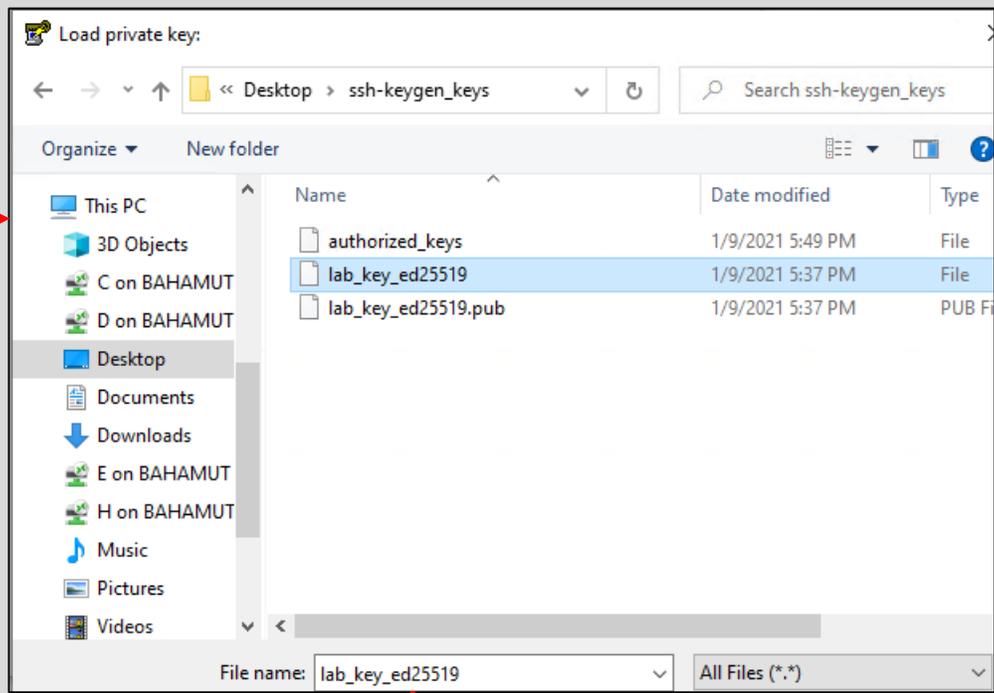
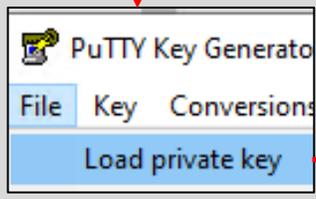
C:\Users\Tony>ssh-keygen -t ed25519 -f D:\Tony\Desktop\ssh-keygen_keys\lab_key_ed25519
Generating public/private ed25519 key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in D:\Tony\Desktop\ssh-keygen_keys\lab_key_ed25519.
Your public key has been saved in D:\Tony\Desktop\ssh-keygen_keys\lab_key_ed25519.pub.
The key fingerprint is:
SHA256:bwGtSHGgPUXctCjHbtFXF5903wJarUQHPQEaZ5Y7DcI tony@StarFall
The key's randomart image is:
+--[ED25519 256]--+
|      +*=oO=oo+ |
|    + =E@++o.+ |
|  . = @.*+o+. |
|    B Bo++ o |
|    S *  ....o |
|      o      . |
+-----[SHA256]-----+
```



Continued to *fig. 15-21*

15-20: Next up, run `ssh-keygen -t ed25519 -f [directory path]/[key name]`. Check out the illustration for an example on how to format the `-f` option. You'll have the option to set a password for the SSH private key if you wish. When `ssh-keygen` is finished running, there should be a `[keyname]` and a `[keyname].pub` file. Using *Notepad++*, edit the `[keyname].pub` file. Ensure the entire file is one line long, and that the End of Line character is set to Unix (LF). Save the modified file as `authorized_keys`. Check out [section 15.2.3.1](#) (pp. 750-759), if you need a refresher on how to do this.

Continued from *fig. 15-20*



15-21: Last but not least, open up puttygen, and select *File > Load private key* from the navigation menu, then navigate to the location of your [keyname] file Make sure. If you password protected the key you generated from running *ssh-keygen*, you'll need to enter that password now. With the private key imported, click the *Save private key* button under the *Actions* section, and save the .ppk file.

15.2.3.2 Copying the SSH public key to lab VMs

At this point, students should have a private key .ppk file, and a public key file named, `authorized_keys`. That `authorized_keys` file needs to be copied to a specific directory on the SIEM, IPS, and Kali virtual machines, and with specific file and folder permissions. While there are a multitude of ways to do what I am about to show you, I'm going to demonstrate three of the easiest methods to do this.

Method 1: WinSCP

If students haven't done so already, download, install, then start the WinSCP application. For this demonstration, we will be using the default "commander" view. Upon starting WinSCP, A window labeled *Login* will pop up. The pane on the left features an icon that reads *New Site*, while the left-side of the window will read *Session*, and have a series of input boxes – *Host name*, *Port number*, *User name* and *Password*. Fill out these fields with the following information:

File Protocol: This is a drop-down menu. Ensure that the option *SFTP* is selected. This should be the default setting.

Host name: 172.16.1.3 (or, if students are using an alternative IP address scheme, the IP address assigned to their SIEM VM)

Port number: 22 (this should be the default setting)

User name: username of the user created on the SIEM VM

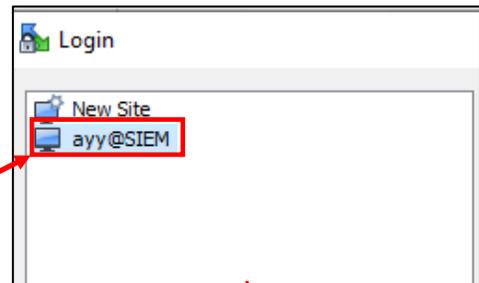
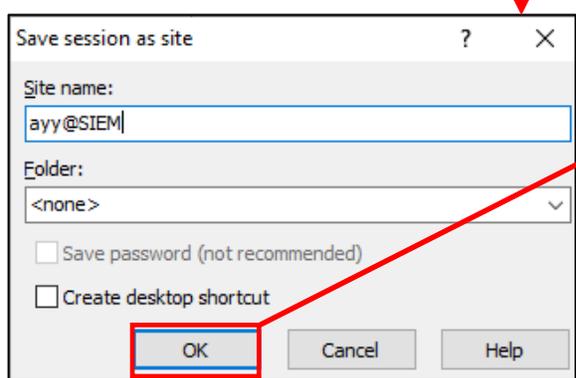
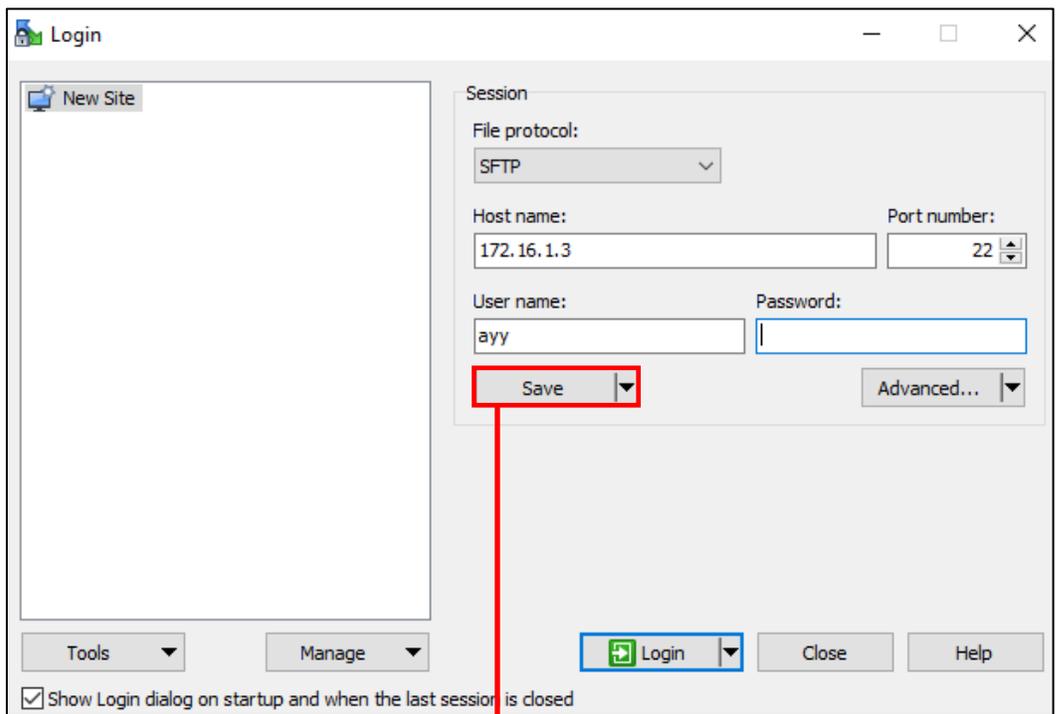
Password: [leave blank]

When finished, click the *Save* button, and a new dialogue box will appear, labeled *Save session as site*. Think of sites as WinSCP's way of creating and saving connection profiles, just like mRemoteNG. By click the *Save* button, students are telling WinSCP they want to save the connection information they entered for later use. The input box labeled *Site name* is the name that will be attached to the connection profile. I would recommend changing this field to something descriptive, such as `[username]@[Virtual Machine Name]`. For example, `ayy@SIEM`. Otherwise, click the *OK* button to proceed. This will cause a new icon featuring the *Site name* students just configured under the *New Site* icon, in the left pane.

Highlight the New site icon, and repeat this process two more times for the IPS and Kali VMs:

Virtual Machine:	IPS	Kali
File Protocol:	SFTP	SFTP
Hostname:	172.16.1.4 (or alternate IP address as necessary)	172.16.2.2 (or alternate IP address as necessary)
Port number:	22	22
User name:	Username created during OS install	Username created during OS install
Password:	Blank	Blank
Site name:	<code>[username]@IPS</code>	<code>[username]@Kali</code>

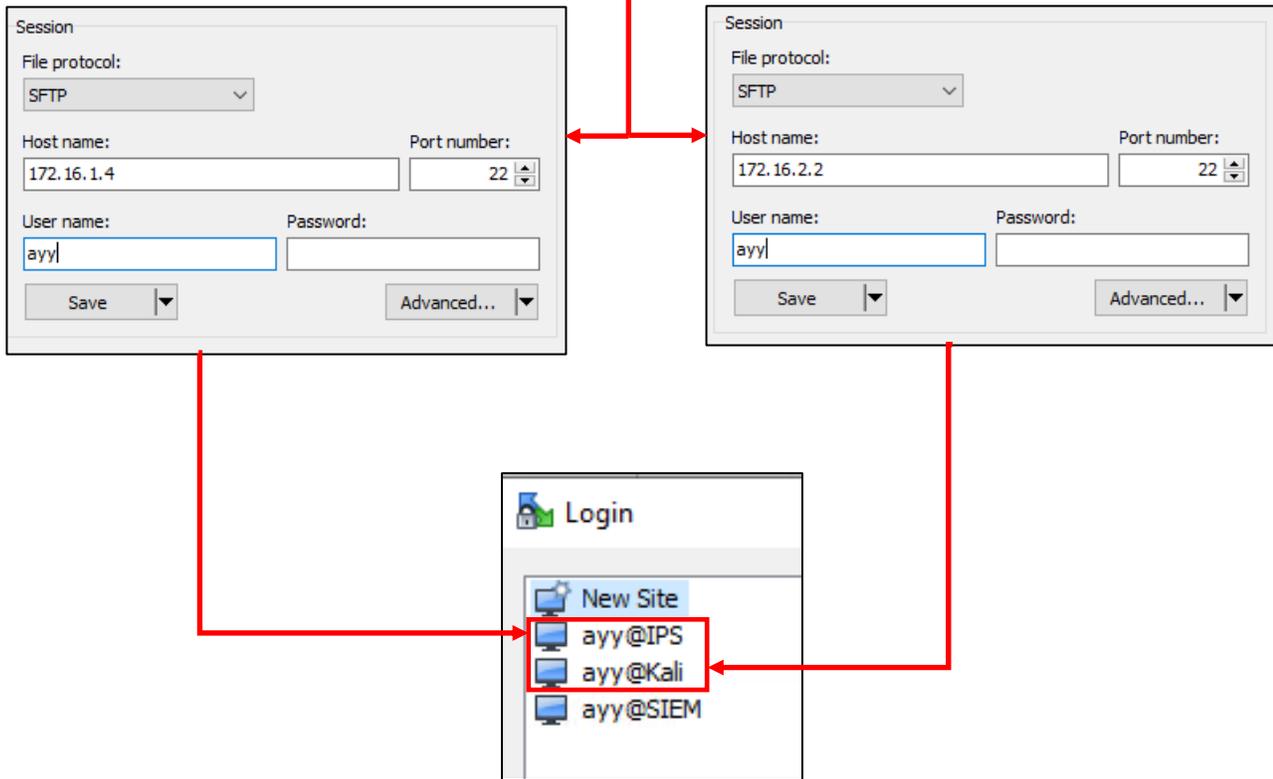
When finished, students should have three site profiles under *New site*, on the *Login* window – one each for the SIEM, IPS, and Kali virtual machines.



Continued to fig. 15-23

15-22: Open the WinSCP application. With the *New Site* icon highlighted in the *Login* window, fill out the fields of the *Session* section of the Window. Select *SFTP* from the *File Protocol* drop-down, enter the IP address of the SIEM VM in the *Host name* input box, verify the *Port number* is set to 22, enter the username created during initial setup into the *User name* input box, then click *Save*. In the *Save session as site* dialogue box, enter a name to describe this connection profile in the *Site name* input box, then click the *OK* button. A new site connection appears in the login window, under the *New Site* icon.

Continued from *fig. 15-22*



15-23: Repeat the process, and create two additional profiles for the IPS and Kali virtual machines. When finished, students should have three site profiles in total.

With all three profiles created, double-click on the site profile for SIEM. Students should be prompted to enter the password for their user on the SIEM VM. After entering the password correctly, the "Commander" interface for WinSCP will appear. There's a lot going on with interface, but don't panic. The only portions of the interface we're concerned with, are the two primary panes in the middle of the screen. Think of the left pane a file browser for the local windows system, while the right pane is a file browser for the remote system (e.g., the SIEM VM). The current directory students are browsing on the local and remote sides of the connection are displayed in a small blue bar above each pane. Students can drag and drop files from one pane to the other in order to transfer files using the SFTP protocol.

The right pane, by default, will display the user's home directory students logged in as. For example, on my SIEM VM, I logged in as the user ayy. That user's home directory is /home/ayy. For this exercise, **do not navigate way from the SIEM user's home directory**. For the left pane, students will need to navigate to the folder the used to store their authorized_keys public file. For example, on my system, I opted to store the public and private keys I wanted to use for my lab in the folder D:\Tony\Desktop\Lab SSH Keys.

In order to change directories, double click on the blue bar above the left pane, and a new window, labeled *Open directory* will appear. Focus on the input box, also labeled *Open directory*. The input box will contain the current directory that WinSCP is displaying on the local Windows system. Students may either manually type in the full directory path to the folder they wish to display, or click the *Browse* button to open windows explorer and select the folder that way. Once you have modified the *Open directory* input box to the directory where the authorized_keys file is located, click the *OK* button at the bottom of the Open directory window to update the WinSCP commander interface, and display the contents of that directory.

The next task is copying the authorized_keys file from the Windows pane to the user's home directory on the SIEM VM. This task is as simple as dragging and dropping the file from the left pane to the right. Once finished, click the tab labeled *New Session* to bring up the *Login* window, and repeat the process of copying the authorized_keys file to the IPS and Kali virtual machines. Once the authorized_keys file has been copied to all three systems, students may exit WinSCP.

SSH Host Keys and You (Part 2)

It's very likely that upon attempting to connect to your virtual machines using WinSCP, that you received one of the following warnings:

Continue connecting to to an unknown server and add its host key to a cache?

Warning - Potential Security Breach The server's key does not match the one WinSCP has in cache.

Not unlike the warnings you got when you attempted to connect to your VMs over SSH using mRemoteNG, these are warnings referring to the SSH host key. The first error message appears if its your first time connecting to the system using WinSCP, and asks users to confirm if they want WinSCP to remember the host and mark it as a known, good host. The second warning occurs if users are connecting to a hostname or IP address that is already known to WinSCP, but its host key doesn't match what it has on file. Again, these errors are very similar to "I don't have this number in my phonebook. Do you want to fix that now?" and "This system's phone number has changed. That could either be very suspicious, or completely benign. Do you want to update the phone number?" In the case of our lab environment, click *Yes* if prompted to continue connecting, or click *Update* to update the existing host key on file, and continue connecting.



WARNING - POTENTIAL SECURITY BREACH!

The server's host key does not match the one WinSCP has in cache. This means that either the server administrator has changed the host key, the server presents different key under certain circumstance, or you have actually connected to another computer pretending to be the server.

The new Ed25519 key details are:

Algorithm: ssh-ed25519 255
SHA-256: bG5NfgfVWpHiBEJ4shxYeYafQm0eMpWcsl7hdMGCJ0s=
MD5: 55:64:18:b6:af:fa:e0:47:e9:10:64:a8:2e:2c:d3:fe

If you were expecting this change, trust the new key and want to continue connecting to the server, either press Update to update cache, or press Add to add the new key to the cache while keeping the old one(s). If you want to carry on connecting but without updating the cache, press Skip. If you want to abandon the connection completely, press Cancel. Pressing Cancel is the ONLY guaranteed safe choice.

[Copy key fingerprints to clipboard](#)



Continue connecting to an unknown server and add its host key to a cache?

The server's host key was not found in the cache. You have no guarantee that the server is the computer you think it is.

The server's Ed25519 key details are:

Algorithm: ssh-ed25519 255
SHA-256: /5riNTzUweYNYa+/cT0MRIHnuDzgd9hEip81Ad33sCo=
MD5: 7f:62:78:d7:9c:42:57:7b:00:f8:d4:21:95:df:08:6e

If you trust this host, press Yes. To connect without adding host key to the cache, press No. To abandon the connection press Cancel.

[Copy key fingerprints to clipboard](#)

15-24: Just like when you used mRemoteNG to connect to your virtual machines for the first time, WinSCP wants to make sure you're aware that these contacts haven't been added to your address book, and/or the phone number for this contact has changed, you should consider updating the phone number.

The final thing students must do is log in to their SIEM virtual machine over SSH using mRemoteNG, and run a series of commands:

```
pwd
ls -al
chmod 600 authorized_keys
mkdir ~/.ssh
chmod 700 ~/.ssh
mv authorized_keys ~/.ssh/authorized_keys
ls -al ~/.ssh
```

In total, there are seven commands. The first command, `pwd`, shows users the current directory they are located in. By default, when logging in, students should be in their user's home directory – in my case, this is `/home/ayy`, and the `pwd` (print working directory) command confirms this.

Next up, is `ls -al`. This command lists the files in the directory specified. If no directory is specified in the command, it displays the files in the current directory. The options passed to `ls` instruct it to show all hidden files (`-a`) and show the files in long list format (`-l`) telling us about the file permissions of these files. With this command, we're looking to make sure that the `authorized_keys` file is present, and looking to see if a hidden directory `".ssh"` is already present in the user's home directory. By default, the `.ssh` directory should not be here, but if for some reason it is already present on the virtual machine, then the `mkdir ~/.ssh` command will fail. We'll talk more about that in just a moment.

Next is the `chmod` command. `chmod` is responsible for modifying permissions to files or folders. To make a very long story short, every file in a Linux/Unix environment has a set of three permissions: Read (4), Write (2), and Execute (1). These three permission sets apply to the user, group, and world (think of world permissions as anyone else that is not the user or group that owns the file, attempting to access the resource on the system). `chmod` represents these three permissions and the three entities as three digits, ranging from 0 through 7 (There are other special permissions and ways to represent file permissions that I'm glossing over here, but they don't apply to what we're attempting to accomplish here today). Students will use `chmod` twice. The first time around, students apply the permission set `600` to the file `authorized_keys`. This means that the only the user that owns the file has read (4) and write (2) access to the file ($2 + 4 = 6$). Any users in the group the file belongs to, and all other users on the system have their permissions set to 0, so they have no access to the file at all. We'll be talking about the second run of this command in a moment.

Note: In the previous section, we used WinSCP and logged in as the user `ayy`. We then transferred the file `authorized_keys` file to the SIEM VM. How did the system decide that user `ayy` is the file's owner? It's sort of a built-in feature of file transfer protocols: Whichever user you happen to use to transfer files on to a given system becomes the owner of those files in almost all cases. As always, there are special cases, but for the time being, just go with this explanation for now.

That brings us to the `mkdir` command. This command is responsible for creating directories and subdirectories. For students used to Windows terminology, directories are another word for folders. In our string of commands above, we are specifically running `mkdir ~/.ssh`. What does this do?

First and foremost, on Linux/Unix systems, there is a special shortcut that can be used to refer to the current user's home directory. This shortcut is the combination "`~`". Next, remember with the `ls -a` command how I referred to the `-a` function as listing all files in a directory, even hidden files and folders? By default, there are a bunch of hidden files and folders in a user's home directory. These hidden files and folders are usually configuration files, folders, and environment files that are used by a bunch of different applications on a system. Hidden files and folders usually have a period (`.`) in front of their name. So, students are asking the `mkdir` command to create the hidden directory, `.ssh`, in the currently logged in user's home directory (`~`).

If while reviewing the output from the `ls -a` command students noticed there's already a `.ssh` directory in the user's home directory, then do not run the `mkdir` command. If you do it anyway, you'll get the error:

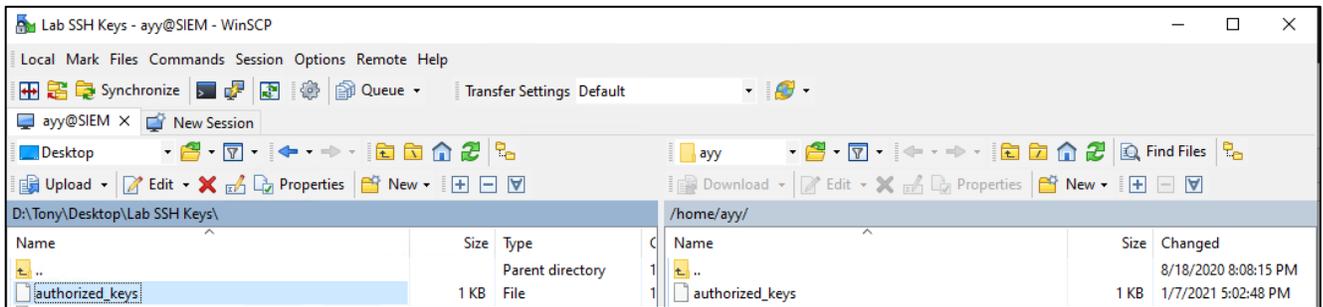
```
mkdir: cannot create directory '/home/[username]/.ssh': File exists
```

This is just a notification from the `mkdir` that the directory is already present and that it will not overwrite the existing `.ssh` directory. Proceed as though the command ran successfully.

Next, students will run the `chmod` command again, this time applying `700` permissions (read (4) + write (2) + execute (1) = 7) to the new folder `~/.ssh`. This effectively configures the folder to where only the user that owns the folder can access or modify the files contained within.

Next, students will use the `mv` command. This command is used to move files from one directory to another, or can be used to rename files. In this instance, the command `mv authorized_keys ~/.ssh/authorized_keys` will relocate the `authorized_keys` file located in the current directory (e.g. `/home/[username]`) to the `~/.ssh` directory, with the exact same file name.

Finally, students utilize the command `ls -a ~/.ssh` to produce a list of files located in the `.ssh` directory in the current user's home directory. This command is run in order to confirm that the `authorized_keys` file was moved to the `.ssh` directory successfully. Once students have confirmed that the file has been placed in the directory successfully, type `exit` to leave the `ssh` session on the SIEM VM, then repeat this process on the IPS and Kali VMs.



```

ayy@siem:~$ pwd
/home/ayy
ayy@siem:~$ ls -al
total 32
drwxr-xr-x 3 ayy ayy 4096 Jan 14 18:15 .
drwxr-xr-x 3 root root 4096 Aug 19 00:08 ..
-rw-rw-r-- 1 ayy ayy 101 Jan 7 22:02 authorized_keys
-rw----- 1 ayy ayy 245 Jan 7 02:28 .bash_history
-rw-r--r-- 1 ayy ayy 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ayy ayy 3771 Feb 25 2020 .bashrc
drwx----- 2 ayy ayy 4096 Aug 19 00:15 .cache
-rw-r--r-- 1 ayy ayy 807 Feb 25 2020 .profile
-rw-r--r-- 1 ayy ayy 0 Aug 19 00:16 .sudo_as_admin_successful
ayy@siem:~$ chmod 600 authorized_keys
ayy@siem:~$ mkdir ~/.ssh
ayy@siem:~$ chmod 700 ~/.ssh
ayy@siem:~$ mv authorized_keys ~/.ssh/authorized_keys
ayy@siem:~$ ls -al ~/.ssh
total 12
drwx----- 2 ayy ayy 4096 Jan 14 21:02 .
drwxr-xr-x 4 ayy ayy 4096 Jan 14 21:02 ..
-rw----- 1 ayy ayy 101 Jan 7 22:02 authorized_keys

```

15-25: After transferring the `authorized_keys` file to the SIEM, IPS and Kali VMs, students will then need to run a series of commands on all three virtual machines to ensure the `authorized_keys` file is in the correct location with the proper file permissions configured:

```

pwd
ls -al
chmod 600 authorized_keys
mkdir ~/.ssh
chmod 700 ~/.ssh
mv authorized_keys ~/.ssh/authorized_keys
ls -al ~/.ssh

```

The output from these commands should be practically identical to what is displayed in the screen capture above. **Just to re-iterate, these commands must be run on the SIEM, IPS and Kali virtual machines.**

Method 2: Copy and Paste

This method of transferring the authorized keys file to the SIEM, IPS, and Kali VMs is more or less about the same, with but a single twist: Instead of using WinSCP to copy the file, then moving it into the correct location with the correct file permissions, this time around, students will make use of Notepad++ and the echo command to copy the contents of the file from the Windows host to the target system directly over an SSH session. Let's get started.

First things first, open *Notepad++* and open the `authorized_keys` file created earlier in this chapter. A neat feature about Notepad++ is that even once a file is saved, Notepad++ will store that file in a tab that can automatically be accessed the next time the application is open. However, if the `authorized_keys` file is not already displayed, select *File* from the Navigation menu, then *Open*. Browse to the location of the `authorized_keys` file, and open it.

Open mRemoteNG and open an SSH session to the SIEM virtual machine, and run the following commands:

```
pwd
ls -al
mkdir ~/.ssh
chmod 700 ~/.ssh
```

Since we already went over the purpose of these commands in-depth while discussing method 1, here is a very brief summary of what we're doing: `pwd` to confirm we're in the SIEM user's home directory. `ls -al` to see if the `.ssh` directory already exists. If it doesn't exist, `mkdir ~/.ssh` to create it. If it does exist, students can skip the `mkdir` command. Finally, the `chmod` command is for setting file permissions to where only the user who owns the folder can access the `~/.ssh` folder. Once students have finished logging into the SIEM VM, and inputting these commands, bring up your *Notepad++* session.

With the `authorized_keys` file opened, and the first (and only) line in the file selected, right click on the line, and select the option *Select All* from the context menu. This will highlight all of the contents of the file. Right click on the line again, and this time, select *Copy* from the context menu. Go back to the SSH session to the SIEM VM in mRemoteNG, and run the following command:

```
echo "[right click once. Only. Once.]" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
ls -al ~/.ssh
```

The `echo` command is normally used to display output to the terminal. In most cases, it is used in shell scripts to display output to users. A special quirk of PuTTY and mRemoteNG is that if you have text copied to the Windows clipboard, right-clicking on a terminal session is the same as pasting that text on to the command line (fun fact: Linux/macOS users can "middle click" on their mouse to do more or less the same thing). When students right-click, they are dumping ALL of the contents of the `authorized_keys` file on the Windows desktop to the command line of the

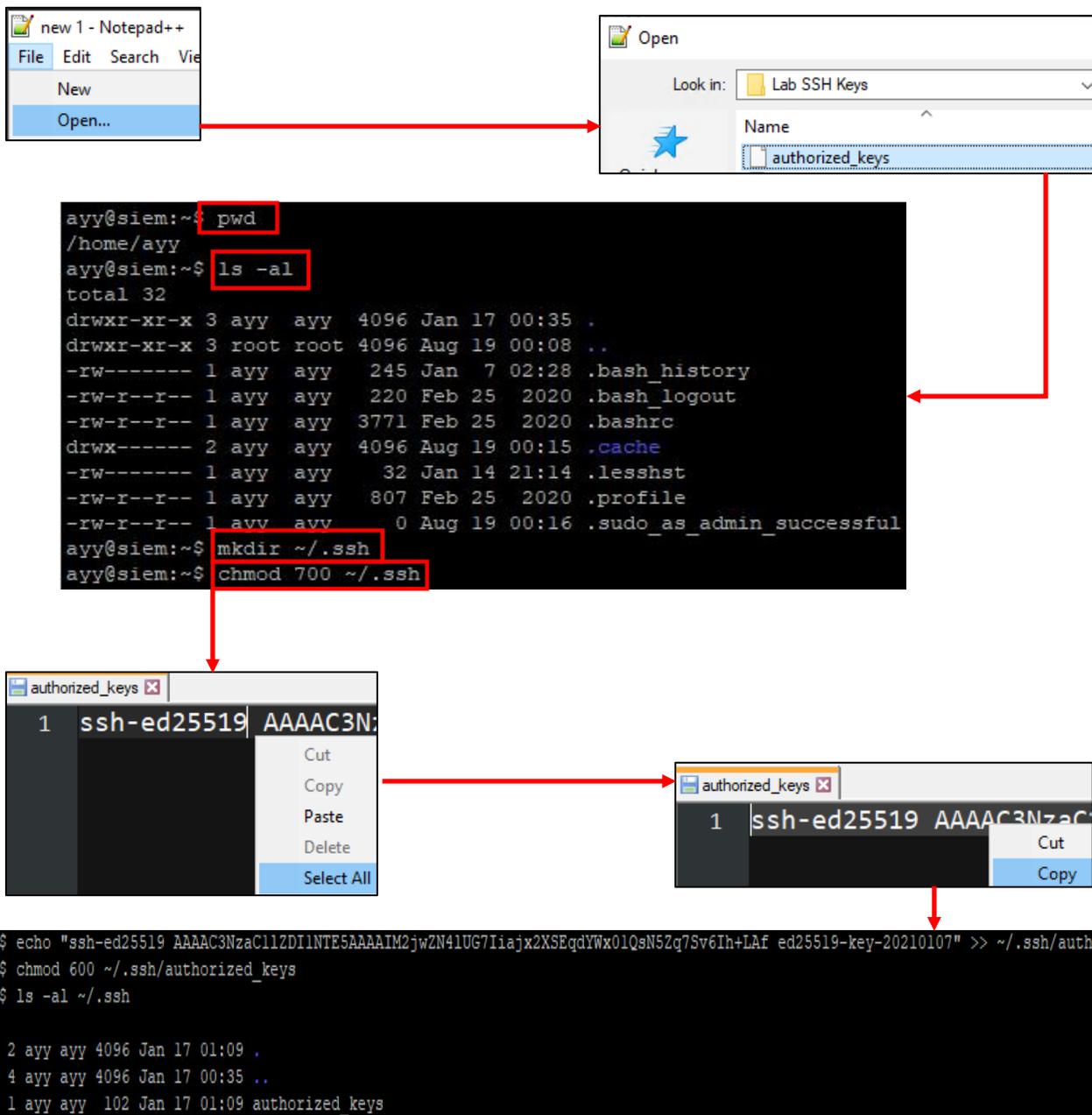
SSH session on SIEM. Immediately after the contents of the key (encased in quotation marks) are the symbols , >>. These characters are used to redirect output from a command. In particular, these characters tell the command "Take the output from this echo command and write it to a file I specify. If the file doesn't exist create it. If a file already exists by that name in the location, append the output of this command to that file." Finally, students instruct the command to write the output to the file ~/.ssh/authorized_keys. As an example, this is what the command looked like with my SSH public key:

```
echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM2jwZN4lUG7Iiajx2XSEqdYwx01QsN5Zq7Sv6Ih+LAFed25519-key-20210107" >> ~/.ssh/authorized_keys
```

This command is followed by `chmod 600 ~/.ssh/authorized_keys` in order to restrict access to the newly created file to where, once again, only the owner of the file can read or write to it. Finally, students run `ls -al ~/.ssh` to verify that the `authorized_keys` file exists in the correct location.

Now that students have followed this process once in the SIEM VM, repeat the process two more times on the IPS and Kali VMs. Here is a brief summary of the tasks to perform:

- Establish SSH sessions to the IPS and Kali VMs
- Create the ~/.ssh directory, and configure the proper file permissions (700)
- Open the authorized_keys file in Notepad++. Using the right-click context menu *choose Select All*, followed by *Copy*
- Back in the SSH sessions to the IPS and Kali virtual machines, use the echo command specified above and use right-click to paste in the contents of the authorized_keys file
- Once students have redirected the output of the echo command to the file ~/.ssh/authorized_keys, set the correct file permissions (600)



15-26: Method 2 is a little bit different from Method 1. This method involves taking advantage of copy/paste and file redirection. Open *Notepad++* and load the `authorized_keys` file we created together earlier in this chapter. Next, open mRemoteNG, and open an SSH session to the SIEM VM. Verify you are in the SIEM user's home directory using `pwd`. Use the `ls -al` command to see if the `.ssh` directory exists in the user's home directory. If its not there, run `mkdir ~/.ssh` followed by `chmod 700 ~/.ssh` to create the directory and set the correct file permissions. Next, switch back to Notepad++, and using the right-click menu choose *Select All*, then select *Copy*. Go back the SSH session on the SIEM VM, and using the echo command examples provided on pages 765 and 766 paste the contents of your `authorized_keys` file into the terminal so that their contents can be written into the `~/.ssh/authorized_keys` file. Finally, use the command `chmod 600 ~/.ssh/authorized_keys` to set the correct file permissions for this file. Afterwards, repeat this process on the IPS and Kali virtual machines.

Method 3: I hope you're not afraid of vi

In a nutshell, the third method I will be demonstrating for students today is practically identical to the second method, the only difference is that this method uses the (in)famous text editor, vi. I've saved this method for last, only because it was covered in the previous revision of this book, and because the text editor vi isn't very intuitive or easy to use at first.

Students will perform more or less the exact same actions as they performed for method 2:

- Open up *Notepad++*, load up the `authorized_keys` file
- Open an SSH session to the SIEM VM and run the following commands:
 - `pwd`
 - `ls -al`
 - `mkdir ~/.ssh` (if the `.ssh` directory doesn't exist)
 - `chmod 700 ~/.ssh`
- Switch back to the *Notepad++* document. Use the right click context menu, and choose *Select All*, followed by *Copy* in the `authorized_keys` file to store the contents in the file in the clipboard

Here is where things go a little bit differently. Switch back to the SSH session to the SIEM VM in mRemoteNG. Run the command:

```
vi ~/.ssh/authorized_keys
```

This opens the vi text editor. Follow these instructions exactly, and we'll be out of here in no time:

- Press the 'i' key on the keyboard to enter vi's insert mode. This is the mode that allows users to input text to a file. Students will know when they are in insert mode, because bottom left corner of the editor will read `-- INSERT --`
- Immediately right click in the SSH session a single time in order to paste in the contents into the file. **Make sure that only one line of content has been pasted into the file.**
- Hit the escape key (esc) on the keyboard to exit insert mode, then immediately type `:wq!` to write the public key data to the `~/.ssh/authorized_keys` file, and exit vi immediately.

Note: If for some reason vi is not behaving properly, don't be afraid to exit and start over. Hit the escape (esc) key a couple of times, then type: `q!` to immediately exit without saving any of your changes.

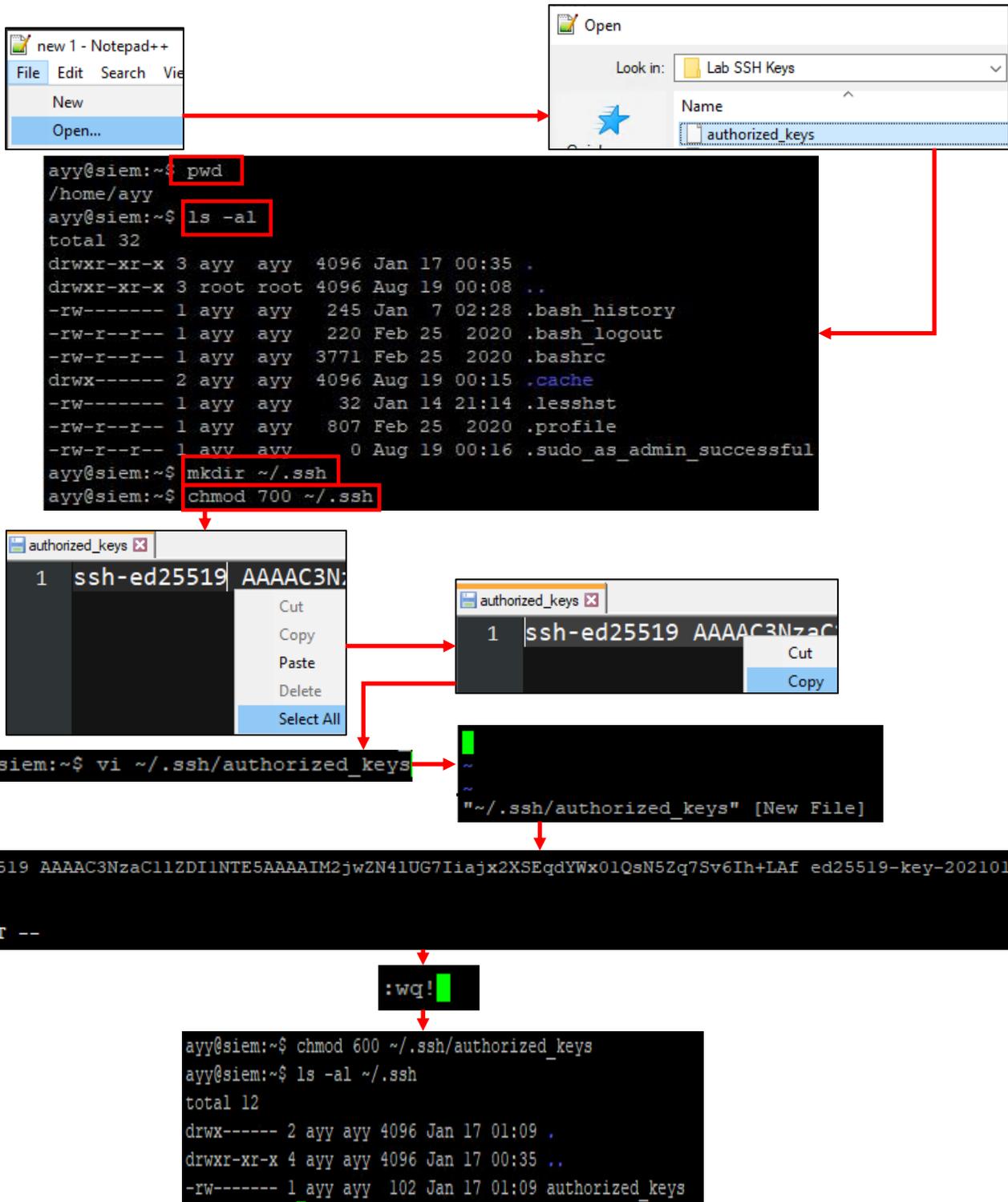
As with method 2, run the commands `chmod 600 ~/.ssh/authorized_keys`, followed by `ls -al ~/.ssh` to set the correct file permissions for the `authorized_keys` file. Repeated this process for the IPS VM, however for the Kali VM, read the sidebar below, *vi'ing Harder*.

'vi'ing Harder

For some reason or another, Kali's vi package doesn't treat right clicking in the editor over mRemoteNG the same way as Ubuntu. Instead, right-clicking in vi on Kali will put you into VISUAL mode, which does absolutely nothing for us. To paste over the contents of the `authorized_keys` file over to the Kali VM, perform the following tasks:

- With the contents of the `authorized_keys` file copied to the Windows clipboard, run `vi ~/.ssh/authorized_keys`
- Immediately enter the key combination `shift + insert (ins)` on your keyboard. The contents of the `authorized_keys` file should be pasted into vi without even entering insert mode. Immediately type `:wq!` To save the file `~/.ssh/authorized_keys`, and exit vi.

This is a special shortcut for PuTTY-based terminal emulators (like PuTTYNG, used by mRemoteNG) that can be used to paste the contents of the clipboard, when right-click pasting isn't working properly.



15-27: The first half of this illustration is identical to method 2. The twist is using vi to write the contents of the `authorized_keys` file to `~/.ssh/authorized_keys`, instead of using `echo`, and output redirection. Students run `vi ~/.ssh/authorized_keys`, hit the 'i' key to enter insert mode, right-click paste the contents of the clipboard, then hit the escape key (esc), followed by typing `:wq!` to write the `authorized_keys` file data. Then, just like with method 2, students run `chmod 600 ~/.ssh/authorized_keys` to set the correct file permissions, then `ls -al ~/.ssh`. Afterwards, this process can be repeated on the IPS VM, but for the Kali VM, students will need to read the sidebar conversation *vi'ing harder* to deal with a quirk on Kali Linux that hinders this method a little bit.

15.2.3.3 Reconfiguring mRemoteNG to Use SSH keys

By this point, students should have an SSH private key, and an SSH public key named `authorized_keys`, in a specially formatted file. A copy of this file should be present in the directory `~/.ssh` on the SIEM, IPS and Kali VMs. The next step is configuring mRemoteNG to use the SSH keys for authentication, as opposed to a password.

Open mRemoteNG. Select *Tools* from the Navigation menu, then select *Options*. In the *mRemoteNG Options* window, select the option labeled *Advanced*, then click the *Launch PuTTY* button. A new window labeled *PuTTYNG Configuration* appears.

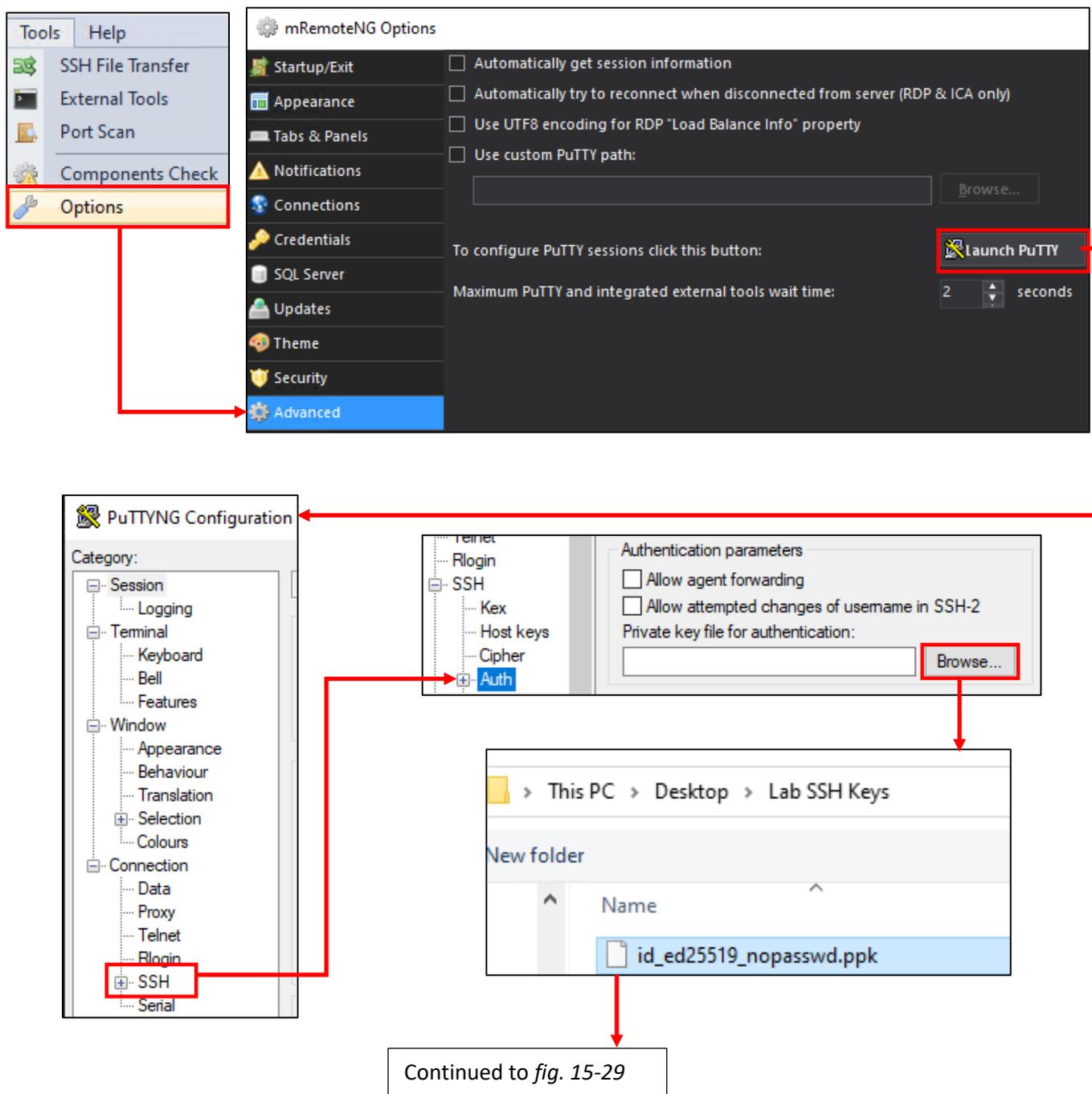
On the left side of the window is a pane labeled *Category*. Look for the option labeled *SSH* and click the small "+" icon next to it. This causes additional options related to SSH connections to appear. Find the option labeled *Auth*, and left-click to highlight it. The right portion of the windows changes. Scroll down to the section labeled *Authentication parameters*. Click the *Browse* button under the option labeled *Private key file for authentication*. An explorer window will appear. Browse to the location of the `.ppk` file we created with puttygen earlier in this chapter, and select it. The input box changes to reflect the full path to the `.ppk` file students just selected.

Back in the *Category* pane, click the option labeled *Session*. Once again, the configuration options on the right will change. In the *Saved Sessions* input box, located in section *labeled Load, save, or delete a stored session*, input a name for the PuTTY session. I would recommend something like "SSH_key_auth(no_password)" if students are using a private key with no password, or "SSH_key_auth(password)", if using a password protected private key. Afterwards, click the *Save* button, and your session will appear in the pane below the input box, with the default settings session. Once finished, click the *Close* button in the bottom right corner of the *PuTTYNG Configuration* window, then click the *OK* button in the *mRemoteNG Options* window.

Now comes the important part. All of those configuration changes we just made together were to create a putty session profile that knows where our private key is. The next step is editing the mRemoteNG connection profiles to use the new PuTTY session. We'll start by modifying the SIEM connection profile. Open mRemoteNG, and click on the SIEM connection profile, under the *Connections* pane. This will cause the *Config* pane to show students all of the configuration data associated with the SIEM VM's connection profile. Students will need to perform the following tasks:

- Under the *Connection* portion, delete the password from the connection profile entirely
- Under the *Protocol* portion, click on the third field, labeled PuTTY Session, then click on the downward facing arrow to review a drop-down menu. Click on the name of the PuTTY session we created together moments ago
- Double-click on the SIEM connection profile. If everything was done correctly, students should get output similar to [fig. 15-17](#) (no password set on SSH private) or [fig. 15-18](#) (password set on SSH private key) on pages 754-755

After confirming that the SSH session to the SIEM VM is working with key-based authentication enabled, perform the same configuration changes to the IPS and Kali SSH connection profiles, then test to confirm they work as well. If all goes well, then congratulations! Students are more or less done here. If things don't go well, see [section 15.4, Troubleshooting SSH Connectivity and Key-Based Authentication](#), pp. 809-812. Alternatively, if you're really having a hard time with key-based authentication and would rather just skip it, see the sidebar conversation, *What if I don't want to use 2FA with mRemoteNG*.



Continued to *fig. 15-29*

15-28: Now comes the fun part, where we tell mRemoteNG to use the SSH keys we generated and moved all around earlier. In mRemoteNG, Select *Tools > Options*, and in the *mRemoteNG Options* menu, select *Advanced*, and click *Launch PuTTY*. In the Category pane, click the "+" next to *SSH*, then left-click on the *Auth* text. Under the section labeled *Authentication parameters*, click *Browse*, and locate the private key .ppk file we generated together earlier in this chapter, and select it.

Continued from *fig. 15-28*

Private key file for authentication:
D:\Tony\Desktop\Lab SSH Keys\id_ed:

Category:

- Session
- Logging
- Terminal
 - Keyboard
 - Bell
 - Features
- Window
 - Appearance
 - Behaviour
 - Translation
 - Selection
 - Colours
- Connection
 - Data
 - Proxy
 - Telnet
 - Rlogin
 - SSH

Basic options for your PuTTYNG session

Specify the destination you want to connect to

Host Name (or IP address) Port

Connection type:
 Raw Telnet Rlogin SSH Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Load, save or delete a stored session

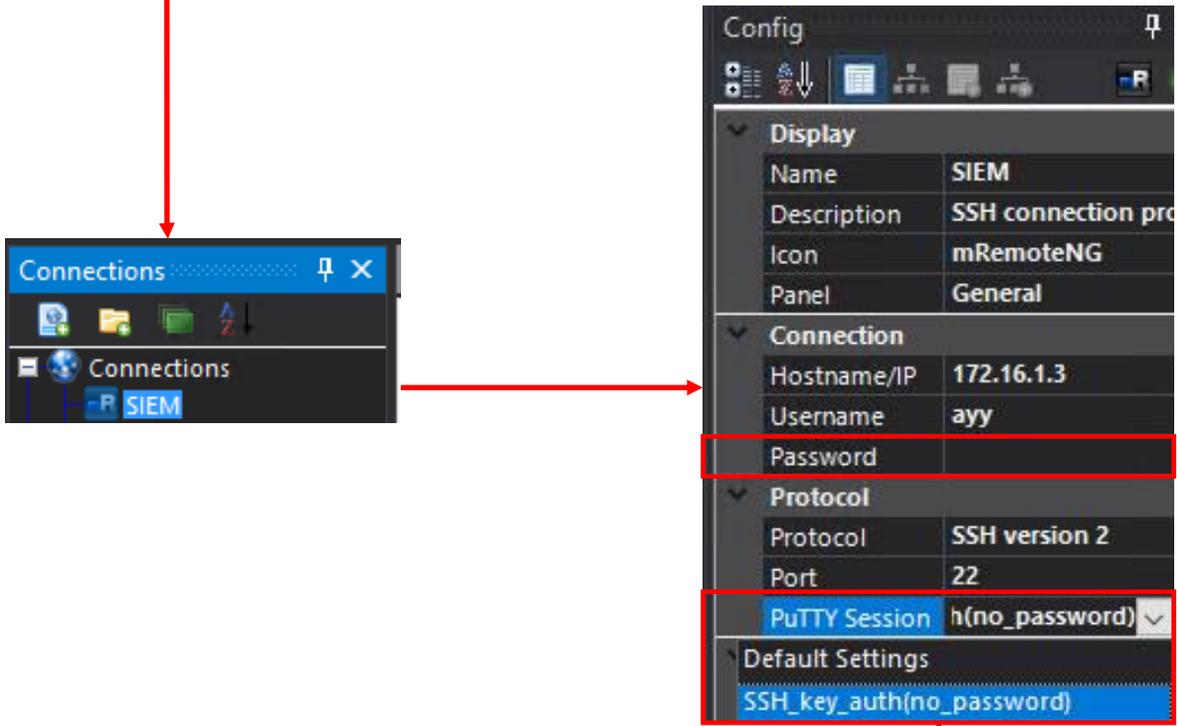
Saved Sessions

Default Settings

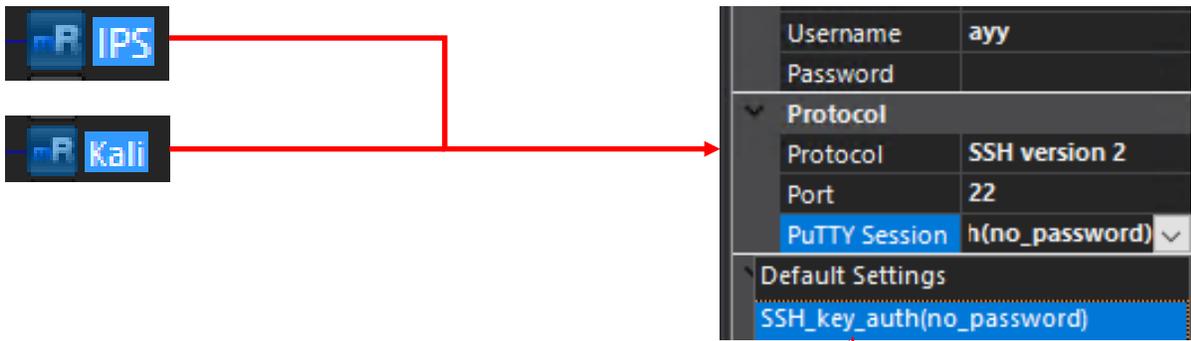
Continued to *fig. 15-30*

15-29: After locating and selecting the .ppk file, click on *Session* under the *Category* pane. In the *Saved Sessions* input box, name your custom PuTTY session, then click the *Save* button. – For example, I choose `SSH_key_auth(no_passwd)` to denote my SSH key is not password protected, and will not provide two-factor authentication. Your saved session will appear in the pane below the input box, along with the session named *Default Settings*. Click the *Close* button to exit the PuTTYNG settings window, then click *OK* to close the *mRemoteNG Options* window.

Continued from *fig. 15-29*



```
Using username "ayy".  
Authenticating with public key "ed25519-key-20210107"
```



```
Using username "ayy".  
Authenticating with public key "ed25519-key-20210107"
```

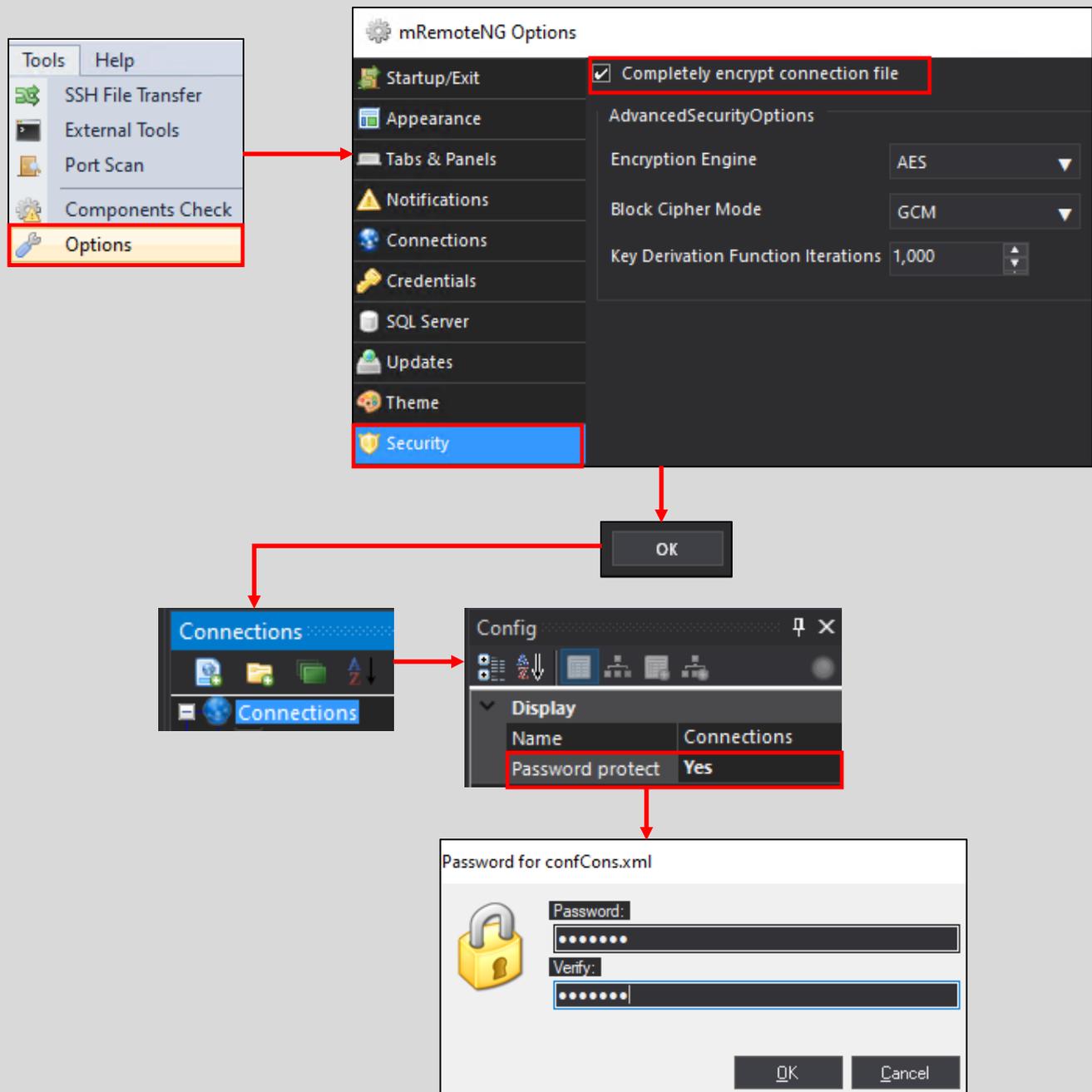
15-30: Highlight the SIEM Connection profile, and in its *Config* window, under *Connection*, delete the contents of the password field. Then, under *Protocol*, click *PuTTY Session*, and select the putty session students just created from the drop-down menu that appears. When finished, double click on the SIEM connection profile and confirm that you get an SSH session either without entering a password at all, or by entering the password of the SSH private key (if students decided to set one). Repeat this process for the IPS and Kali virtual machines, removing the stored password, and changing their PuTTY Session configuration as well.

What if I don't want to use 2FA with mRemoteNG?

If you're having a hard time with getting key-based authentication up and running, even after attempting to troubleshoot it, just remember that this guidance is here to try and make your lab life easier! Key-based auth is meant to either enable laziness (no password protected key), or enhance security (2FA – password and key). If you don't plan on using two-factor authentication with mRemoteNG for whatever reason, here are my recommendations:

-In general (but especially if you have to share your hypervisor host system with others), **avoid saving your passwords in mRemoteNG**. mRemoteNG stores all of your connection data (connection profiles, username, password, protocol, etc.) to a file named `confcons.xml`. This file is usually stored in `C:\Users\[username]\AppData\Roaming\mRemoteNG`. By default, the file is not encrypted. There are ways to fix this, but in general, if you can avoid it, you should really store and retrieve your passwords from a password manager instead.

- If you enjoy the convenience of double-clicking that connection profile and not having to futz around with your password manager, and are willing to accept the risks that come with it, there are things you can do to help secure the `confcons.xml` file. First, visit the *mRemoteNG Options* menu (under *Tools > Options* in the navigation menu), Then click on the option labeled *Security*. In the security options menu, check the *Completely encrypt connection file* checkbox, then click *OK* to exit. Back in the main window, click on the globe labeled *Connections*, in the *Connections* pane. In the *Config* pane down below, there is an option labeled *Password protect*. Highlight it, click the downward facing arrow to open a drop-down menu, and change the default of *No* to *Yes*. This immediately causes a dialogue box to pop up, labeled *Password for confCons.xml*. Enter a password into the *Password* and *Verify* input boxes to set a password for your connection file. **Do not lose this password! It will be required to open the connection file every time you open mRemoteNG!** In fact, you might want to store it in a password manager.



15-31: If you're having a hard time with setting up key-based auth and don't want to bother trying to figure it out, you can make mRemoteNG memorize the password to your virtual machines (like we've been doing so far in this chapter), but there are risks that come with this. If someone else uses your computer, all they have to do is obtain a copy of your `confCons.xml` file to grab credentials to all connections managed by mRemoteNG – by default, this file is not encrypted.

The good news is that this behavior can be changed by checking the *Completely encrypt connection file* checkbox, located under *Security* in the *mRemoteNG Options* menu (*Tools > Options*). After clicking *OK* to exit the options menu, click the *Connections* listing under the *Connections* pane, then in the *Config* panel below, set the *Password protect* option to *Yes*. This immediately prompts users to set a password for the `confCons.xml` file. ***Do not lose or forget this password!*** You will need to enter it every time you start mRemoteNG in order to be able to access your connection profiles. In fact, consider saving it in a password manager.

Bonus Lesson: Key-Based Authentication with WinSCP

Way back in Chapter 1, students were advised to have the application WinSCP installed on their system. In fact, one of the (easier) methods for transferring the `authorized_keys` file for SSH key-based authentication had us utilize WinSCP. In the coming chapters (and perhaps, later when and if you choose to customize your lab environment to better suit your needs), you may need to transfer additional files to or from your workstation/hypervisor host to your lab virtual machines. WinSCP will be extremely useful for performing that task quickly and effectively.

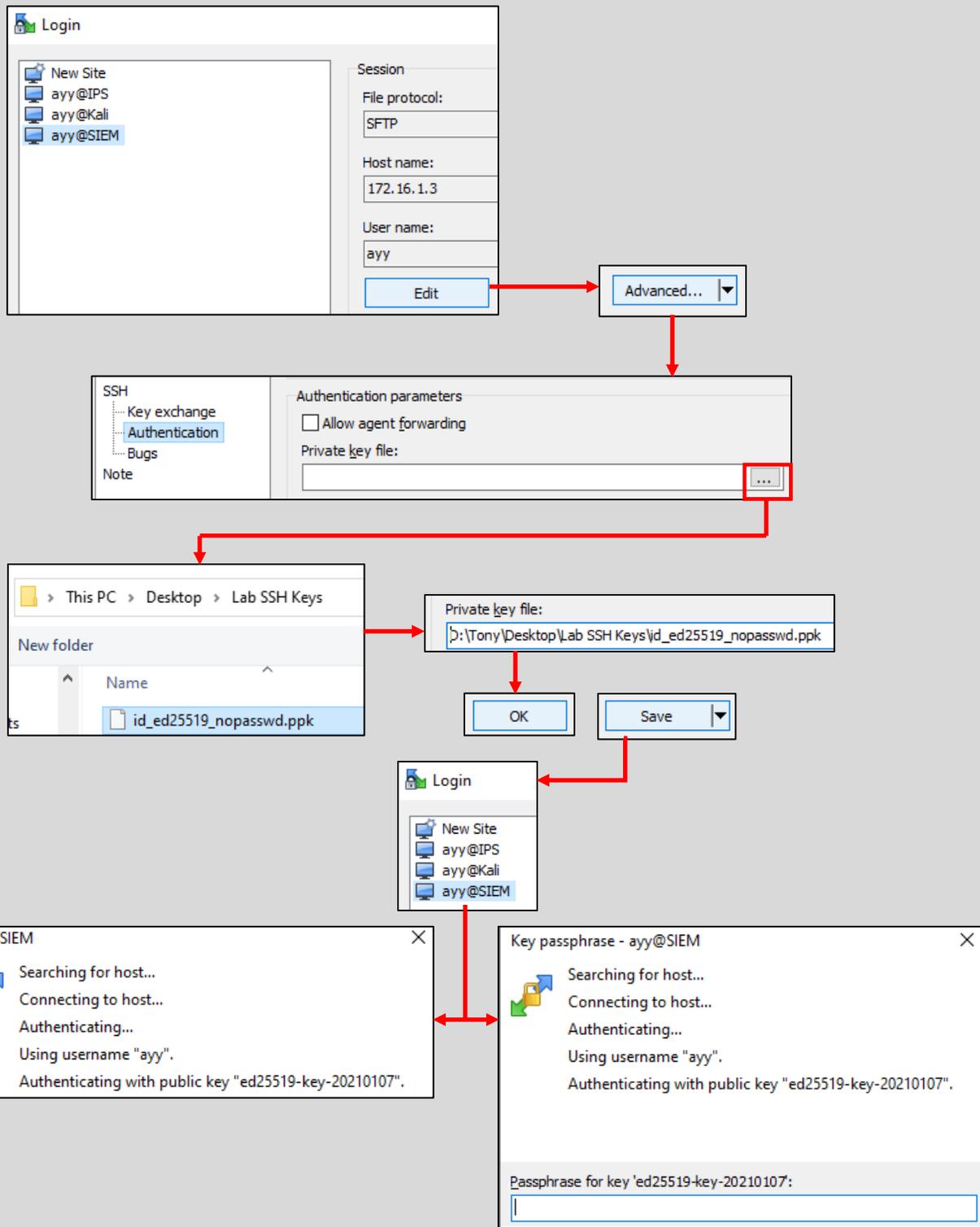
Conveniently, WinSCP supports key-based authentication for its connection profiles, just like mRemoteNG. Also conveniently, it wants SSH private key in `.ppk` format. Please note that the instructions below assume that you have already performed the steps necessary to configure SSH key-based authentication for the SIEM, IPS and Kali virtual machines and that it is working properly.

Open the WinSCP application. If you followed the process for using Method 1 to transfer your `authorized_keys` file to the SIEM, IPS, and Kali VMs, you should already have connection profiles configured for all three VMs. Check out [pp. 760-768](#) if you need guidance on how to perform this task. We'll begin by editing the SIEM connection profile. In the *Login* window, select the SIEM connection profile on the left pane, then click the *Edit* button on the right portion of the window, in the *Session* section. Next, click the button labeled *Advanced*, and a new window labeled *Advanced Site Settings* will appear.

In the pane on the left, click on the option labeled *Authentication*, under *SSH*, and a variety of options will appear on the right side of the window. We're interested in the input box labeled *Private key file*, under the *Authentication parameters* section. Click the grey button (labeled ...) on the far right in the input box to open a file browser. Browse to the location of the `.ppk` file we generated earlier using *puttygen*, and double click on it to select it. The input box will update to reflect the full file path to private key. Click the *OK* button below to close the *Advanced Site Settings* window. Back in the *Login* window, click the *Save* button to save the changes we've made to the SIEM connection profile. Repeat this process with the IPS and Kali connection profiles.

When finished, test out your new connection profiles to make sure that key-based authentication is working properly. Double click on the SIEM connection profile, and confirm you are able to get an SCP session either without a password (if you chose not to password protect your private key file) or with the password used to encrypt your private key file. Again, repeat this process on the IPS and Kali connection profiles to verify key-based authentication is working as intended.

Please be advised that later in this book, students will need to transfer some installer files to some of the lab virtual machines, and that SCP is going to be all but required to do that. **If you decide to disable password authentication over SSH on your lab virtual machines later in this chapter (Section 15.5.2), WinSCP will no longer work without configuring key-based authentication.**



15-32: Open WinSCP, select the SIEM connection profile, then click *Edit*, followed by *Advanced*. On the *Advanced Site Settings* screen, click on *Authentication*. In the *Private key file* input box, click the grey button with the ellipsis (...) and browse to the location of your .ppk file. When finished, click *OK*, then back in the *Login* window, click *Save*. Repeat this process for the IPS and Kali profiles. Once finished, double click on the SIEM connection profile and test to confirm that key-based authentication is working as intended. Depending on whether or not you password protected your private key, you may either be connected automatically, or prompted for the password to decrypt your private key, if everything is working properly. Again, test the IPS and Kali connection profiles as well.

15.3 Remote Access for Linux/MacOS Hypervisor Hosts

In this section, students will learn how to configure remote access to their lab virtual machines hosted on MacOS or Linux, using the `ssh` command. Students will also learn how to generate SSH keys, and configure their virtual machines for key-based authentication. This will allow students to choose between instantly getting a session on their VMs, or configuring two-factor authentication using a password-protected SSH key. MacOS, and most Linux distributions have the `ssh` client, and several utilities available by default. If you are unsure if your system has `ssh` utilities installed, refer back to Chapter 1, [section 1.6](#), pp. 27-28 for further guidance.

Note: In addition to confirming that there's an SSH client available on your MacOS or Linux hypervisor host, students must also confirm that they have a valid route to the OPT1/IPS network (typically, 172.16.2.0/24), and that the `ssh` service on the Kali Linux VM is enabled and running in order to accept SSH connections.

Static routes on MacOS and Linux were covered in sections [15.1.2](#), [15.1.3](#), and [15.1.3.1](#) (pp. 733-741). Enabling and starting the `ssh` service on the Kali Linux VM was covered in section [15.1.4](#), p. 742.

15.3.1 The `ssh` command

Connecting to virtual machines using the SSH protocol is considerably easier on Linux and MacOS systems than it is on Windows systems (though, admittedly Windows has been getting better about this in recent years). The `ssh` command, in its most basic form, has the following format:

```
ssh [user]@[hostname]
```

As an example, let's assume students wish to connect to the SIEM VM at 172.16.1.3, using the username `ayy`. That command looks like:

```
ssh ayy@172.16.1.3
```

Likewise, the command to attempt to connect to the IPS VM (172.16.1.4) and Kali VM (172.16.2.2) would look like:

```
ssh ayy@172.16.1.4  
ssh ayy@172.16.2.2
```

Run these commands, substituting the username and assigned IP addresses of the SIEM, IPS and Kali virtual machines as necessary. Students will be prompted for the password of the user for each virtual machine. When the correct password is supplied, students should be logged in to the virtual machine of choice remotely, using the SSH protocol. Please note that upon first connection, the `ssh` client may prompt students about the authenticity of the SSH host key, asking if they wish to continue connecting. To make a long story short, respond `yes` to continue. For more details, check out sidebar conversation, *SSH Host Keys, but MacOS/Linux* for more details.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ayy@172.16.1.3
ayy@172.16.1.3's password:
ayy@siem:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ayy@172.16.1.4
ayy@172.16.1.4's password:
ayy@ips:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ayy@172.16.2.2
ayy@172.16.2.2's password:
ayy@kali:~$
```

15-33: Use the ssh command to connect to the SIEM, IPS and Kali virtual machines, using the username, passwords, and IP addresses assigned when creating the virtual machines. Upon first connecting, students may be prompted to accept the SSH host key for their virtual machines. See the sidebar conversation, *SSH Host Keys, but MacOS/Linux* for more details.

SSH Host Keys, but MacOS/Linux

Some of you might have noticed that when you connect to your virtual machines that you might get security warnings similar to what is displayed in *fig. 15-34*.

One version of this message will read:

```
The authenticity of host 'x.x.x.x (x.x.x.x)' can't be established.
```

While another version of this message reads:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
```

```
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
```

```
It is also possible that a host key has just been changed.
```

The SSH protocol utilizes a feature called host key fingerprinting. Think of this as a way of positively identifying the server you are trying to connect to via SSH to help counter Man-in-the-Middle and/or Spoofing attacks.

The first version of this message warns users and tells them "Hey, there is an SSH server running on the IP address you want to connect to, but I don't have its host key on file. Are you sure you want to continue?" If you select yes, then the host key gets added to a file in your current user's home directory called `known_hosts` (`~/.ssh/known_hosts`). Think of this as a phone book the computer can use to verify the identify of systems you want to connect to.

The second message warns users, "There is an SSH server running on this host, but its host key fingerprint doesn't match the fingerprint I have on file." This can happen when you create a new virtual machine and it inherits the same IP address as a previous virtual machine whose SSH host key was already on file, so now the SSH client is convinced the new virtual machine is an imposter.

The primary difference between the ssh client on Linux/MacOS and mRemoteNG on Windows is that you just can't click OK to tell ssh to update the host key for you automatically. The easiest way to fix this problem is to find the offending line in the `known_hosts` file, and delete it. Then, when you try to connect to the host in the future, you'll get the first message, asking if you want to add the host key to the `known_hosts` file, instead of flat out refusing to connect. The fastest way to do this is through the `ssh-keygen` command – in particular its `-R` option has the ability to remove host keys from the `known_hosts` file for you:

```
ssh-keygen -R [hostname/ip address]
```

For example, if I needed to remove the hostkey for IP address 172.16.1.3:

```
ssh-keygen -R 172.16.1.3
```

After removing the old host key for the host giving you trouble, use the SSH command to connect to it again, and you'll get the warning that the host fingerprint's authenticity cannot be established, and the prompt to add it to the known_hosts file (e.g. the first error message).

```
trobenson@trobinsons-MacBook-Pro ~ % ssh ayy@10.0.0.149
The authenticity of host '10.0.0.149 (10.0.0.149)' can't be established.
ECDSA key fingerprint is SHA256:sB8RtGB9gan/PFdlUz8yuDlbyQ4QXSr3vv+V7WQEFUA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? █
```

```
trobenson@trobinsons-MacBook-Pro ~ % ssh ayy@172.16.1.3
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
```



```
trobenson@trobinsons-MacBook-Pro ~ % ssh-keygen -R 172.16.1.3
# Host 172.16.1.3 found: line 16
/Users/trobenson/.ssh/known_hosts updated.
Original contents retained as /Users/trobenson/.ssh/known_hosts.old
```

15-34: Think of SSH host keys, and the known_hosts file, as a phonebook for SSH servers that your system's ssh client recognizes and trusts. The first image depicts a notification that gets displayed when you attempt to connect to a new server via SSH – it's just asking to confirm that you want to consider this system trusted, and continue connecting. The second, more ominous -looking notification occurs if you're like me, and you add, remove, and re-create systems and virtual machines constantly, causing the ssh host key for that hostname/IP address to change. The difference being that if you get the second error message, the ssh client will flat out refuse to allow you to connect, until you either correct the ssh host key in the known_hosts file, or remove it entirely. Fortunately, ssh-keygen -R [hostname/ip address] can be used to remove the specific entry out of the known_hosts file, without having to use a text editor, or worry about mangling the file.

15.3.2 Connection profiles and ~/.ssh/config

Now that students have a feel for the basics of the `ssh` command, how can we make it easier to connect to each virtual machine? Windows users get connection profiles and mRemoteNG, where connections can be made just by double-clicking an icon, so what do the Linux and MacOS users get? The `~/.ssh/config` file.

Note: "~/" is a Linux/MacOS shortcut that means "the current user's home directory". For example, if I were logged in as the user `ayy` on a Linux system, "~/" would refer to `/home/ayy`. So, `~/.ssh/config` is the exact same as `/home/ayy/.ssh/config`.

The `ssh config` file, located in a hidden directory named `.ssh` in the user's home directory, allows users to define host aliases for connecting to remote servers over the SSH protocol. What does this mean? If configured properly, all students need to do is type `ssh siem`, and the SSH client knows the username to connect as, and the IP address of the SIEM virtual machine. Supply the password for the pre-configured user, and that's it.

In this section, I will help students generate their own SSH config one of two ways: Manually, or by downloading a copy I have hosted on gist.github.com. But before we get into that, let's talk about the structure of this file.

While there are a lot of configuration options and directives you can pass to the SSH config file, there only three directives that we need to focus on: `Host`, `HostName`, and `User`. The `Host` field defines the name you want to give to the system you are connecting to. In my example above, the `Host` directive is set to `siem`. `HostName` on the other hand refers to either the IP address or domain name of the system to connect to. In my lab environment, the IP address configured for the SIEM VM is `172.16.1.3`, so that would be the value for the `HostName` field. Finally, we have the `User` field. As the name implies, the `User` field defines what user to specify when connecting to a particular host. The user I created when installed Ubuntu Server to the SIEM vm was `ayy`, so that is the value I would specify for the `User` field. Now, it's one thing to know these values and how to set them, but it's another to know how the config file expects these values to be formatted. So, let's do a little demonstration:

```
Host ayylmao
    HostName 127.0.0.1
    User ayy

Host xcom
    HostName skyranger.local
    User menace15
```

In this example, I specified two hosts, `ayylmao`, and `xcom`. For the Host `ayylmao` profile, I specified `127.0.0.1` as the `HostName` to connect to, and the username `ayy` in the `User` field. So, if I were to run the command:

```
ssh ayylmao
```

It is the equivalent of:

```
ssh ayy@127.0.0.1
```

For Host `xcom`, I specified `skyranger.local` as the `HostName` to connect to. So the system will use DNS to attempt resolve `skyranger.local` to an IP address before connecting. For the `User` field, I specified the username `menace15`. So, if I were to run the command:

```
ssh xcom
```

That is the equivalent of:

```
ssh menace15@skyranger.local
```

Notice that both entries start with the `Host` field, and that the fields that came after for both entries (e.g., `HostName` and `User`) were both tab indented. Finally, notice that there was a single blank line between the `Host ayylmao` and `Host xcom` entries.

Frontloading Work Now to Enable Laziness Later

This looks like a lot of work for very little gain, but the `ssh config` file supports a lot more options and directives than what I have demonstrated so far. In chapter 16, we'll be talking about making a `config` entry to enable multiple Local and Dynamic TCP forwarding listeners to enable SSH tunneling and using an SSH session as a sort of proxy. Not only that, later on in this chapter, when we learn to configure key-based authentication, one of the methods used for enabling that involves using the `scp` command to copy a file to the SIEM, IPS and Kali virtual machines. The `scp` command benefits from the profiles configured in the SSH `config` file. For example:

```
scp ~/authorized_keys ayylmao:~/
```

Is the same as:

```
scp ~/authorized_keys ayy@127.0.0.1:~/
```

If you want to know more about the tons of configuration options the `config` file supports, run the command `man ssh_config` on either Linux or MacOS. Alternatively, a wonderful resource I found on the internet about this subject was from the blog post:

<https://linuxize.com/post/using-the-ssh-config-file/>

and here is a link using the internet archive's wayback machine:

<https://web.archive.org/web/20210101210331/https://linuxize.com/post/using-the-ssh-config-file/>

Now that students understand the configuration options and layout of the `~/.ssh/config` file, the next task is to create one of their own for the lab environment. Either copy the lines below manually into a text file, or copy the complete file from:

<https://gist.github.com/da667/9610bf9f39aa5906f2d5c13e136911b7>

For students copying the contents displayed here, remember to match the formatting, and ensure that Unix line feeds (LF) are configured for the file, using your preferred text editor.

ssh config file

```
Host siem
    Hostname 172.16.1.3
    User ayy

Host ips
    HostName 172.16.1.4
    User ayy

Host kali
    HostName 172.16.2.2
    User ayy
```

Note: I'm assuming that you are not naming the user account on all three virtual machines `ayy`, so make sure to modify the `User` setting to reflect the actual user you wish to log in as. Additionally, if students are running their lab environment in which the IP addresses for the SIEM, IPS, and kali virtual machines are not 172.16.1.3, 172.16.1.4, and 172.16.2.2 respectively, modify the contents of the `HostName` field to reflect the actual IP addresses of your lab virtual machines.

At this point, students should have a file named `config` on their system. Now, students will need to move the file to the correct location with the proper file permissions. Begin by running this command:

```
ls -al ~/.ssh
```

The `ls` command is used to list the contents of directory. In this case we're telling the `ls` command to show us all of the files (`-a`) in a "long list" format (`-l`) in the `~/.ssh` directory. that shows us all sorts of details. But really, we don't care about of any of that. The only thing we're interested in is confirming that the `~/.ssh` directory exists. If the `ls` command runs without errors, then that means the directory exists. If students get the error `No such file or directory`, they'll need to create the directory, and set its file permissions using these commands:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
```

`mkdir` tells our system to create the directory, while `chmod` is used to change the access permissions to where only the user you are logged in as has access to that newly created directory. Next, run these commands:

```
cp config ~/.ssh/config
chmod 600 ~/.ssh/config
```

The first command, `cp`, makes a copy of the `config` file you either edited, or downloaded from github, and places it into the `~/.ssh` directory. Be aware that this command assumes `cp` is being ran from the directory where the `config` file is currently located. If not, students will need to specify the complete path to the file. For example, lets assume the `config` file being copied is in `~/Downloads`. Here is what the modified `cp` command would look like:

```
cp ~/Downloads/config ~/.ssh/config
```

The second command, `chmod`, is there once again to ensure that only the current user has access to the `config` file once it is placed in the `~/.ssh` directory.

With the file in the correct directory with the correct file permissions run the following commands:

```
ssh siem
ssh ips
ssh kali
```

If immediately prompted with a password, then the `ssh` command was able to read from the `config` file, and connect to the IP address and username mapped to that host entry. I would recommend supplying the password for your users on all three virtual machines, then run the following commands:

```
hostname
whoami
ip -br a
```

These commands will verify the SSH session has been established with the proper user, to the proper host.

```
trobenson@trobinsons-MacBook-Pro ~ % ls -al ~/.ssh
total 56
drwx----- 7 trobinson  staff  224 Feb  8 15:33 .
drwxr-xr-x+ 26 trobinson  staff  832 Feb  8 15:33 ..
```

```
ayy@potatobox:~$ ls -al ~/.ssh
ls: cannot access '/home/ayy/.ssh': No such file or directory
ayy@potatobox:~$ mkdir ~/.ssh
ayy@potatobox:~$ chmod 700 ~/.ssh
```

```
cp config ~/.ssh/config
chmod 600 ~/.ssh/config
```

```
trobenson@trobinsons-MacBook-Pro ~ % ssh siem
ayy@172.16.1.3's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$ hostname
siem
ayy@siem:~$ whoami
ayy
ayy@siem:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8      ::1/128
enp0s3            UP                172.16.1.3/24   fe80::a00:27ff:feb9:caa4/64
```

```
trobenson@trobinsons-MacBook-Pro ~ % ssh ips
ayy@172.16.1.4's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$ hostname
ips
ayy@ips:~$ whoami
ayy
ayy@ips:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8      ::1/128
enp0s3            UP                172.16.1.4/24   fe80::a00:27ff:feb6:c63f/64
enp0s8            DOWN
enp0s9            DOWN
```

```
trobenson@trobinsons-MacBook-Pro ~ % ssh kali
ayy@172.16.2.2's password:
ayy@kali:~$ hostname
kali
ayy@kali:~$ whoami
ayy
ayy@kali:~$ ip -br a
lo                UNKNOWN          127.0.0.1/8      ::1/128
eth0              UP                172.16.2.2/24   fe80::9eee:ab16:1657:leaf/64
```

15-35: Run `ls -al ~/.ssh` to confirm whether or not your hypervisor host already has that directory present. If it is not present, use `mkdir` to create it, and `chmod` to set the correct permissions to the directory. Create the SSH config file using the information provided on the previous page. Copy it to `~/.ssh/config`, run `chmod` on the file, then run `ssh siem`, `ssh ips`, and `ssh kali`. Students should run the `hostname`, `whoami`, and `ip -br a` commands to confirm successful login to the intended VM.

What if I'm already using a ~/.ssh/config?

I'm going to assume that if you already have a ~/.ssh/config file present on your MacOS or Linux hypervisor host, that you're proficient enough to know how to add entries to the file manually. However, if you don't share my confidence, try the following:

- Download or create the config file as instructed in this section – either download it, or use a text editor to create a copy with Unix line feeds.

-Run the commands:

```
cp ~/.ssh/config ~/.ssh/config.old
cat config >> ~/.ssh/config
```

The first command will back up the existing SSH config file to a file named config.old in the ~/.ssh directory. If you run into a problem, you can then use cp ~/.ssh/config.old ~/.ssh/config to restore the old copy of the file before we mucked about with it. The second command runs the cat command against the config file you edited. Again, if you are not currently in the same directory as your newly edited config file, the command will need to change to reflect that. The cat command displays the content of text files to the terminal. The ">>" redirects and **appends** that output to the destination specified – in this case, ~/.ssh/config.

After appending this information to your existing SSH config file, test it like we did above. Confirm that the connection profiles are working properly. As a side note, if your SSH config file contains an entry that is similar to this:

```
Host *
    User jbradford
```

Note that the SSH config file sort of operates in the order of the profiles configured, and that they sort of compound with one another. This means that, if you appended the config file we made together in this chapter to a config file that has these lines in it, the ssh command will assume that for every single host, you want to specify the user jbradford. The easiest way to fix this would be to edit the ~/.ssh/config file and add this profile to the bottom of the file, like so:

```
Host siem
    Hostname 172.16.1.3
    User ayy

Host ips
    HostName 172.16.1.4
    User ayy

Host kali
    HostName 172.16.2.2
    User ayy

Host *
    User jbradford
```

Alternative Method: The `alias` Command

The `alias` command is a pretty neat Unix/Linux tool. `alias` allows you to define a string or a word as a complex command or a complex series of commands. Many Unix/Linux-based operating systems have a host of aliases defined by default, and these can be viewed by running `alias` with no arguments. Let's look at an example alias I created for use on my macbook:

```
alias keepalive='while true; do w; sleep 10; done'
```

To begin, we run `alias`, and define the name `keepalive`. Then after the equal sign, the commands to run are encased in single quotes (`'`). This alias is multiple commands in a `do/while` loop. Specifically, it's an infinite loop – one that does not have an exit condition. The loop executes the `w` command (displays logged on users), then the `sleep 10` command, pausing execution of the loop for 10 seconds. Since the condition of the loop is set to `true`, the loop never terminates. That means this alias will continuously run the `w` command, followed by sleeping for 10 seconds continuously, unless the loop is ended by some other means (e.g., the user hits `ctrl+c` to cancel execution, or kills the process some other way.)

Some SSH servers and/or clients are configured to drop SSH connections after certain amount of idle time. Rather than trying to futz around with the SSH server or client settings, this is a fast and lazy way to keep an SSH connection open while I'm busy doing other things. All I do is type `keepalive` in the terminal, the alias runs the `while/do` loop that runs as long as I need to, and when I want to regain control of that SSH session, I just use the keyboard combination `ctrl+c` to stop the loop. Can you see where we might be able to use the `alias` command instead `~/.ssh/config` for defining SSH sessions? On your Linux or MacOS hypervisor host, type in these three commands, exactly as displayed:

```
alias siem='ssh ayy@172.16.1.3'  
alias ips='ssh ayy@172.16.1.4'  
alias kali='ssh ayy@172.16.2.2'
```

Replace the IP addresses and usernames as necessary in the aliases above to reflect your lab environment. Next, test the aliases you just created. Type in `siem`, `ips`, and `kali` in the terminal as though they were commands one-by-one to test whether or not the alias opens an SSH session to each virtual machine. You can use the `hostname`, `whoami`, and `ip -br a` commands to verify you are on the correct host, as the correct user like we did for the SSH config file setup.

```
trobinson@trobinsons-MacBook-Pro ~ % alias keepalive
keepalive='while true; do w; sleep 10; done'
trobinson@trobinsons-MacBook-Pro ~ % keepalive
12:56 up 78 days, 22:23, 3 users, load averages: 1.39 1.48 1.46
USER      TTY      FROM            LOGIN@   IDLE   WHAT
trobinson console  -              22Nov20 78days -
trobinson s000    -              28Dec20 12:40 -zsh
trobinson s001    10.0.0.3      10:29      - w
```

```
trobinson@trobinsons-MacBook-Pro ~ % alias siem='ssh ayy@172.16.1.3'
trobinson@trobinsons-MacBook-Pro ~ % alias ips='ssh ayy@172.16.1.4'
trobinson@trobinsons-MacBook-Pro ~ % alias kali='ssh ayy@172.16.2.2'
```

```
trobinson@trobinsons-MacBook-Pro ~ % siem
ayy@172.16.1.3's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$ hostname
siem
ayy@siem:~$ whoami
ayy
ayy@siem:~$ ip -br a
lo                UNKNOWN        127.0.0.1/8    ::1/128
enp0s3            UP              172.16.1.3/24 fe80::a00:27ff:feb9:caa4/64
```

```
trobinson@trobinsons-MacBook-Pro ~ % ips
ayy@172.16.1.4's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$ hostname
ips
ayy@ips:~$ whoami
ayy
ayy@ips:~$ ip -br a
lo                UNKNOWN        127.0.0.1/8    ::1/128
enp0s3            UP              172.16.1.4/24 fe80::a00:27ff:feb6:c63f/64
enp0s8            DOWN
enp0s9            DOWN
```

```
trobinson@trobinsons-MacBook-Pro ~ % kali
ayy@172.16.2.2's password:
ayy@kali:~$ hostname
kali
ayy@kali:~$ whoami
ayy
ayy@kali:~$ ip -br a
lo                UNKNOWN        127.0.0.1/8    ::1/128
eth0              UP              172.16.2.2/24 fe80::9eee:ab16:1657:leaf/64
```

15-36: Aliases are extremely powerful, enabling tons of laziness for a little bit of front-loaded preparation. Aliases allow users to define a string or word that triggers a complex command, or series of commands. In this case, we can use aliases to define SSH connection strings to the SIEM, IPS, and Kali virtual machines that work just about as well as connection profiles in the SSH config file. However, there are some downsides that need to be addressed.

So that's that, right? Well, not exactly. See, aliases by default, are not persistent. What that means is the terminal session you define an alias on is the only terminal session that alias will work on by default. How do we make it so where your aliases are persistent and will run on any terminal session you start? To answer that, we need to talk a little about the different types of shells, and their resource files.

The command line environment on Linux/MacOS systems is referred to as a "shell". There are many different types of shell environments. The most popular ones today are the bash (Bourne Again Shell), and zsh (Z shell) shells. These shells have special configuration files that are placed in the user's home directory called resource files. These resource files can be used to define special settings and configurations that alter how the shell works for that user. Every time a new shell session is started by the user (e.g., an SSH session, a new terminal window or tab, etc.), that shell reads the resource file in their home directory. We can take advantage of this, and define our aliases in that resource file. First and foremost, you need to determine which shell is being used on your system. The easiest way to do this is by running the command:

```
echo $SHELL
```

This command tells your terminal session to print out the value of the environment variable SHELL. As you might've guessed, this environment variable tells the user what shell is currently being used. If your system is using the bash shell, we'll be focusing on modifying the `~/.bashrc` file. If your system uses zsh, we will focus on modifying the `~/.zshrc` file. Before we begin, I recommend making a backup copy of your current resource file using the `cp` command:

```
cp ~/.bashrc ~/.bashrc.old
cp ~/.zshrc ~/.zshrc.old
```

With that out of the way, the easiest way to add our aliases to the resource file is to use the `echo` command and output redirection to append them. That way, we don't have to deal with text editors the rest of the file. Run these commands if your system is using the bash shell:

```
echo "alias siem='ssh ayy@172.16.1.3'" >> ~/.bashrc
echo "alias ips='ssh ayy@172.16.1.4'" >> ~/.bashrc
echo "alias kali='ssh ayy@172.16.2.2'" >> ~/.bashrc
```

For zsh, use these commands instead:

```
echo "alias siem='ssh ayy@172.16.1.3'" >> ~/.zshrc
echo "alias ips='ssh ayy@172.16.1.4'" >> ~/.zshrc
echo "alias kali='ssh ayy@172.16.2.2'" >> ~/.zshrc
```

To test and see if the aliases are functioning properly, exit the current terminal session by typing `exit`, or closing the terminal application, then re-opening it. Next, test to see if the `siem`, `ips`, and `kali` aliases exist, by using the aliases to establish SSH connections to the SIEM, IPS and Kali virtual machines.

```
trobinson@trobinsons-MacBook-Pro ~ % echo $SHELL
/bin/zsh
trobinson@trobinsons-MacBook-Pro ~ % cp ~/.zshrc ~/.zshrc.old
trobinson@trobinsons-MacBook-Pro ~ % echo "alias siem='ssh ayy@172.16.1.3'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro ~ % echo "alias ips='ssh ayy@172.16.1.4'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro ~ % echo "alias kali='ssh ayy@172.16.2.2'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro ~ % exit
```

```
ayy@potatobox:~$ echo $SHELL
/bin/bash
ayy@potatobox:~$ cp ~/.bashrc ~/.bashrc.old
ayy@potatobox:~$ echo "alias siem='ssh ayy@172.16.1.3'" >> ~/.bashrc
ayy@potatobox:~$ echo "alias ips='ssh ayy@172.16.1.4'" >> ~/.bashrc
ayy@potatobox:~$ echo "alias kali='ssh ayy@172.16.2.2'" >> ~/.bashrc
ayy@potatobox:~$ exit
```

```
trobinson@trobinsons-MacBook-Pro ~ % siem
ayy@172.16.1.3's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```

```
trobinson@trobinsons-MacBook-Pro ~ % ips
ayy@172.16.1.4's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```

```
trobinson@trobinsons-MacBook-Pro ~ % kali
ayy@172.16.2.2's password:
```

15-37: Aliases have one slight problem. By default, they do not persist, and aliases made on one terminal session will not apply to other terminal sessions. We can fix this by defining our aliases for the SIEM, IPS, and Kali virtual machines and saving them to the user's appropriate shell resource file. That way, every time a new terminal session is opened, the shell will read the resource file, and define the SSH connection aliases automatically. The screen captures above show the process for verifying what shell is being ran by reading the SHELL environment variable, backing up the existing shell resource file using the cp command, writing our aliases to the ~/.zshrc or ~/.bashrc file (depending on the shell students are using on their hypervisor host), exiting the current terminal session, then starting a new one to confirm our aliases are defined correctly, and working as intended.

15.3.3 Enabling Key-Based Authentication

The SSH protocol allows for a variety of different ways users can authenticate, or prove that they are the user they are trying to connect as. The most common authentication method is by pairing a username with a password (see section 15.3.1). In this section, we will be learning about key-based authentication for Linux/macOS hosts.

To make a long story short, SSH key-based authentication relies on two pieces of information to authenticate a user: A public key stored on the system you want to log in to over SSH, and a private key that your ssh client (e.g. the ssh command) must have access to. **The private key must be kept safe, and as the name implies, private. If someone else has your private key, they can log in as you.** Depending on how the private key is created, key-based authentication can be used on its own with no password to enable fast and easy remote access to your virtual machines, or the SSH private key can be password protected and require users to enter that password every time it is used to connect, providing a basic form of two-factor authentication (*something you have*, the SSH private key, and *something you know*, the password to decrypt and use that private key).

In the coming sections, students will learn how to generate an SSH public and private key pair using ssh-keygen, then transfer the public key file to the SIEM, IPS and Kali virtual machines using one of a variety of transfer methods. **Please note that for the remainder of this guide, it is assumed that students configured an SSH config file with profiles for the SIEM, IPS, and Kali VMs.** Refer to [section 15.3.2](#) (pp. 788-797) for guidance, if necessary.

15.3.3.1 ssh-keygen

To begin, open a terminal session, and run the command:

```
ssh-keygen -t ed25519
```

This command will generate a public and private SSH key using the ed25519 algorithm. The ssh-keygen command will ask you if you want to assign a password to the private key. That choice is entirely up to you, as we discussed above. Password protecting your key makes your lab VMs a bit more secure by providing basic two-factor authentication, while not password protecting it will make SSH access more convenient. When finished, the ssh-keygen command will provide you with two files in the ~/.ssh directory:

```
id_ed25519
id_ed25519.pub
```

The id_ed25519 file is the private key. It should never leave the ~/.ssh directory. As the name implies, this key must be kept private. If the key is not password protected (or if that password is weak and easy to crack), and someone is able to copy it, they can access any system in which key-based authentication has been configured using that public/private keypair. The

id_ed25519.pub file however is an entirely different story. Students will need to get this file into the ~/.ssh directory for the user on the SIEM, IPS and Kali virtual machines.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/trobinson/.ssh/id_ed25519):
Created directory '/Users/trobinson/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/trobinson/.ssh/id_ed25519.
Your public key has been saved in /Users/trobinson/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Mu3v82YozkLmCFqAo0kT46mo2X4uiKFBFuhyVGnBqr8 trobinson@trobinsons-MacBook-
Pro.local
The key's randomart image is:
+--[ED25519 256]--+
| .   ooo          |
| .+. +          |
|+. =o           |
| =Oo           |
|O* .  o S       |
|O o  o+         |
|+X . = . .     |
|B + o o . . o o |
| .E= .  oooo=.  |
+-----[SHA256]-----+
```

15-38: Students will need to run `ssh-keygen -t ed25519` in order to generate their public and private SSH keys. While running this command, students are given the option of password protecting their SSH private key (`id_ed25519`), but it is not required. Our next task involves copying the file `id_ed25519.pub` to the SIEM, IPS, and Kali virtual machines, and renaming the file `authorized_keys` to enable key-based authentication to the other lab VMs. Fortunately, there are multiple ways to do this.

15.3.3.2 Copying the SSH public key to lab VMs

At this point, students should have a public and private key pair generated by the `ssh-keygen` command. The `id_ed25519.pub` file needs to be transferred to the SIEM, IPS, and Kali virtual machines, renamed `authorized_keys`, and placed into the user's `~/.ssh` directory with specific file and folder permissions. While there are a multitude of ways to do what I am about to show you, I'm going to demonstrate three of the easiest methods to do this.

Method 1: `ssh-copy-id`

`ssh-copy-id` is an amazing utility. This command automates the entire process of transferring SSH public keys to remote systems, so long as the user has credentials and can access the remote system using the SSH protocol. The format for the command is also fairly simple:

```
ssh-copy-id [username]@[ip address or hostname]
```

or, alternatively, if students are using an SSH config file:

```
ssh-copy-id [Host]
```

The command will locate the SSH public key for the user who generated the command, transfer it to the remote system, place it in the `~/.ssh` directory, rename the file `authorized_keys`, and set acceptable file permissions all at once. Run the following commands to transfer the public key to the SIEM, IPS and Kali virtual machines:

```
ssh-copy-id siem
ssh-copy-id ips
ssh-copy-id kali
```

```
trobenson@trobinsons-MacBook-Pro ~ % ssh-copy-id siem
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobenson
/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
ayy@172.16.1.3's password:

Number of key(s) added:          1

Now try logging into the machine, with:  "ssh 'siem'"
and check to make sure that only the key(s) you wanted were added.

trobenson@trobinsons-MacBook-Pro ~ % ssh-copy-id ips
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobenson
/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
ayy@172.16.1.4's password:

Number of key(s) added:          1

Now try logging into the machine, with:  "ssh 'ips'"
and check to make sure that only the key(s) you wanted were added.

trobenson@trobinsons-MacBook-Pro ~ % ssh-copy-id kali
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobenson
/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
ayy@172.16.2.2's password:

Number of key(s) added:          1

Now try logging into the machine, with:  "ssh 'kali'"
and check to make sure that only the key(s) you wanted were added.
```

15-39: `ssh-copy-id` makes setting up key-based authentication much easier. The format for the `ssh-copy-id` command is similar to the `ssh` client itself:

```
ssh-copy-id [username]@[ip address/hostname]
```

The command also supports SSH config file Host designations. This means students can run:

```
ssh-copy-id siem
ssh-copy-id ips
ssh-copy-id kali
```

Method 2: scp

The next best method for students to transfer their public keys to their lab virtual machines is to use the scp command. SCP, or the secure copy protocol can be used to transfer files securely, using the SSH protocol for encryption. Long story short, scp can be use to copy files to just about any system students can access using the SSH protocol. The format of the scp command can get a bit wonky, depending on whether not the user is transferring files to the remote system, or downloading files from the remote system. Here is the general format of the command for transferring files to the remote system:

```
scp [/path/to/local/file] [username]@[ip address/hostname]:[/destination/directory]
```

For example, say I wanted to transfer the file `/opt/avenger.tar.gz` on my local system to `192.168.1.100`, logging in as the user `jkelly`, and copy the file to `/backup` on the remote system, that command would look like this:

```
scp /opt/avenger.tar.gz jkelly@192.168.1.100:/backup
```

The `scp` command will run, and place the `avenger.tar.gz` file in the `/backup` directory. I could also rename the file. Instead of specifying `/backup` as the directory to transfer the file to, I could specify `/backup/avenger_research.tar.gz` instead. The logic for transferring files from a remote target to a local system is similar, except reversed. In keeping with the same example, this command will transfer the `/backup/avenger_research.tar.gz` file from `192.168.1.100` logging in as the user `jkelly`, and transfer the file to `/opt/avenger.tar.gz` on the local system:

```
scp jkelly@192.168.1.100:/backup/avenger_research.tar.gz /opt/avenger.tar.gz
```

The `scp` command also supports the SSH config file, allowing students to use the Host designation. In keeping with our example so far, Let's assume that `192.168.1.100` is specified as Host `proving_grounds`, and that the User is already designated as `jkelly` Here is what those commands would look like. First transferring to the remote system:

```
scp /opt/avenger.tar.gz proving_grounds:/backup/avenger_research.tar.gz
```

And transferring from the remote host:

```
scp proving_grounds:/backup/avenger_research.tar.gz /opt/avenger.tar.gz
```

Run the following commands to transfer the file `~/.ssh/id_ed25519.pub` to the SIEM, IPS, and Kali virtual machines, renaming the file `authorized_keys`:

```
scp ~/.ssh/id_ed25519.pub siem:~/authorized_keys
scp ~/.ssh/id_ed25519.pub ips:~/authorized_keys
scp ~/.ssh/id_ed25519.pub kali:~/authorized_keys
```

Next, students will need to SSH to the SIEM, IPS and Kali virtual machines, and run the following commands on each virtual machine:

```
pwd
ls -al
chmod 600 authorized_keys
mkdir ~/.ssh
chmod 700 ~/.ssh
mv authorized_keys ~/.ssh/authorized_keys
ls -al ~/.ssh
```

In total, there are seven commands. The first command, `pwd`, shows users the current directory they are located in. By default, when logging in, students should be in their user's home directory – in my case, this is `/home/ayy`, and the `pwd` (print working directory) command confirms this.

Next up, is `ls -al`. This command lists the files in the directory specified. If no directory is specified, it displays the files in the current directory. The options passed to `ls` instruct it to show all hidden files (`-a`) and show the files in long list format (`-l`) telling us about the file permissions of these files. With this command, we're looking to make sure that the `authorized_keys` file is present, and looking to see if a hidden directory `".ssh"` is already present in the user's home directory. By default, the `.ssh` directory should not be here, but if for some reason it is already present on the virtual machine, then the `mkdir ~/.ssh` command will fail. We'll talk more about that in just a moment.

Next is the `chmod` command. `chmod` is responsible for setting permission to access a file or folder. To make a very long story short, every file in a Linux/Unix environment has a set of three permissions: Read (4), Write (2), and Execute (1). These three permission sets apply to the user, group, and world (think of world permissions as everyone attempting to access the resource on the system). `chmod` represents these three permissions and the three entities as three digits, ranging from 0 through 7 (there are other special permissions and ways to represent file permissions that I'm glossing over here, but they don't apply to what we're attempting to accomplish here today). Students will use `chmod` twice. The first time around, students apply the permission set `600` to the file `authorized_keys`. This means that the only the user that owns the file has read (4) and write (2) access to the file ($2 + 4 = 6$). Any users in the group the file belongs to, and all other users on the system have their permissions set to 0, so they have no access to the file at all. We'll be talking about the second run of this command in a moment.

Note: Whichever user you happen to use to transfer files on to a given system becomes the owner of those files in almost all cases. This is why we do not have to be concerned with who owns the `authorized_keys` file when it is transferred to the lab virtual machines.

That brings us to the `mkdir` command. This command is responsible for creating directories and subdirectories. For students used to Windows terminology, directories are another word for folders. In our string of commands above, we are specifically running `mkdir ~/.ssh`. What does this do?

Remember with the `ls -a1` command how I referred to the `-a` function as listing all files in a directory, even hidden files and folders? By default, there are a bunch of hidden files and folders in a user's home directory. These hidden files and folders are usually configuration files, folders, and environment files that are used by a bunch of different applications on a system. Hidden files and folders usually have a period (.) in front of their name. So, students are asking the `mkdir` command to create the hidden directory, `.ssh`, in the currently logged in user's home directory (`~/`).

If while reviewing the output from the `ls -a1` command students noticed there's already a `.ssh` directory in the user's home directory, then do not run the `mkdir` command. If you do it anyway, you'll get the error:

```
mkdir: cannot create directory '/home/[username]/.ssh': File exists
```

This is just a notification from the `mkdir` that the directory is already present and that it will not overwrite the existing `.ssh` directory. Proceed as though the command ran successfully.

Next, students will run the `chmod` command again, this time applying `700` permissions (read (4) + write (2) + execute (1) = 7) to the new folder `~/ssh`. This effectively configures the folder to where only the user that owns the folder can access or modify the files contained within.

Next, students will use the `mv` command. This command is used to move files from one directory to another, or can be used to rename files. In this instance, the command `mv authorized_keys ~/.ssh/authorized_keys` will relocate the `authorized_keys` file located in the current directory (e.g. `/home/[username]`) to the `~/ssh` directory, with the exact same file name.

Finally, students utilize the command `ls -a1 ~/.ssh` to produce a list of files located in the `.ssh` directory in the current user's home directory. This command is run in order to confirm that the `authorized_keys` file was moved to the `.ssh` directory successfully. Once students have confirmed that the file has been placed in the directory successfully, type `exit` to leave the `ssh` session on the SIEM VM, then repeat this process on the IPS and Kali VMs.

```
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub siem:~/authorized_keys
ayy@172.16.1.3's password:
id_ed25519.pub                                100% 120    71.8KB/s  00:00
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub ips:~/authorized_keys
ayy@172.16.1.4's password:
id_ed25519.pub                                100% 120    78.6KB/s  00:00
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub kali:~/authorized_keys
ayy@172.16.2.2's password:
id_ed25519.pub                                100% 120    85.7KB/s  00:00
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
ayy@172.16.1.3's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```

```
ayy@siem:~$ pwd
/home/ayy
ayy@siem:~$ ls -al
total 32
drwxr-xr-x 3 ayy ayy 4096 Jan 14 18:15 .
drwxr-xr-x 3 root root 4096 Aug 19 00:08 ..
-rw-rw-r-- 1 ayy ayy 101 Jan 7 22:02 authorized_keys
-rw----- 1 ayy ayy 245 Jan 7 02:28 .bash_history
-rw-r--r-- 1 ayy ayy 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ayy ayy 3771 Feb 25 2020 .bashrc
drwx----- 2 ayy ayy 4096 Aug 19 00:15 .cache
-rw-r--r-- 1 ayy ayy 807 Feb 25 2020 .profile
-rw-r--r-- 1 ayy ayy 0 Aug 19 00:16 .sudo_as_admin_successful
ayy@siem:~$ chmod 600 authorized_keys
ayy@siem:~$ mkdir ~/.ssh
ayy@siem:~$ chmod 700 ~/.ssh
ayy@siem:~$ mv authorized_keys ~/.ssh/authorized_keys
ayy@siem:~$ ls -al ~/.ssh
total 12
drwx----- 2 ayy ayy 4096 Jan 14 21:02 .
drwxr-xr-x 4 ayy ayy 4096 Jan 14 21:02 ..
-rw----- 1 ayy ayy 101 Jan 7 22:02 authorized_keys
```

15-40: After transferring the authorized_keys file to the SIEM, IPS and Kali VMs using the scp command, students will then need to run a series of commands on all three virtual machines to ensure the authorized_keys file is in the correct location with the proper file permissions configured:

```
pwd
ls -al
chmod 600 authorized_keys
mkdir ~/.ssh
chmod 700 ~/.ssh
mv authorized_keys ~/.ssh/authorized_keys
ls -al ~/.ssh
```

The output from these commands should be practically identical to what is displayed in the screen capture above. **Just to re-iterate, these commands must be run on the SIEM, IPS and Kali virtual machines.**

Method 3: Copy and Paste

The third and final method I will cover for transferring the public key to the lab virtual machines involves making use copy and paste, echo, and file redirection. This method will be demonstrated using the SIEM VM. Afterwards, students will have to repeat the process for the IPS and Kali virtual machines on their own. Begin by opening a terminal session on the Unix/Linux hypervisor host, and run

```
ssh siem
```

To establish an SSH session on the SIEM VM. Next, open a second terminal window, or new tab, and run the following command

```
cat ~/.ssh/id_ed25519.pub
```

The `cat` command can be used to display the contents of text files directly to the terminal. Next, using your mouse, highlight all of that text, right click on it, and select *Copy*. This will save the text of the public key file to the local system's clipboard. Switch back to terminal session with the SSH session to the SIEM VM, and run the following command:

```
echo "[right click -> paste]" >> ~/authorized_keys
```

The `echo` command is normally used to display output to the terminal. In most cases, it is used in shell scripts to display output to users. When students right-click and select *Paste*, they are dumping ALL of the contents of the `authorized_keys` file from their local Linux/macOS system, to the command line of the SSH session on SIEM. Immediately after the contents of the key (encased in quotation marks) are the symbols, `>>`. These characters are used to redirect output from a command. In particular, these characters tell the command "Take the output from this `echo` command and write it to a file I specify. If the file doesn't exist create it. If a file already exists by that name in the location, append the output of this command to that file." Finally, students instruct the command to write the output to the file `~/authorized_keys`. As an example, this is what the command looked like with my SSH public key:

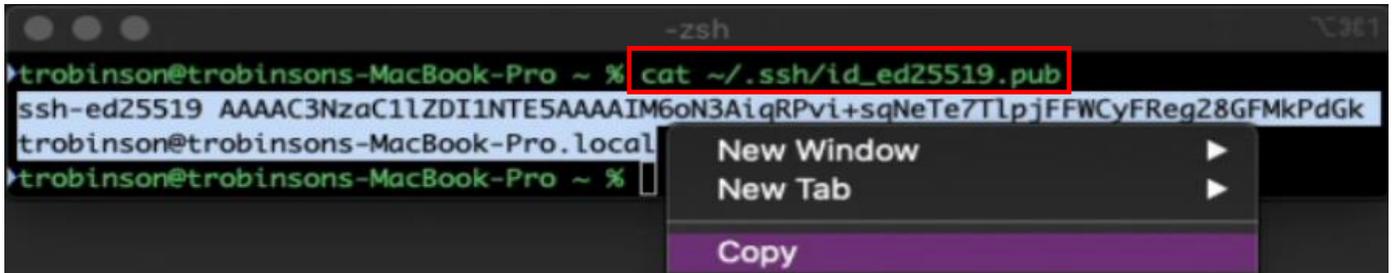
```
echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM6oN3AiqRPvi+sqNeTe7T1pjFFWCyFReg28GFMkPdGk trobinson@trobinsons-MacBook-Pro.local" >> ~/authorized_keys
```

From here, the process of getting the public key into the correct location with the correct file and folder permissions is identical to the process used in method 2. Run these commands:

```
pwd
ls -al
chmod 600 authorized_keys
mkdir ~/.ssh
chmod 700 ~/.ssh
mv authorized_keys ~/.ssh/authorized_keys
ls -al ~/.ssh
```

Repeat this process for the IPS and Kali virtual machines.

```
trobenson@trobinsons-MacBook-Pro ~ % ssh siem  
ayy@172.16.1.3's password:  
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```



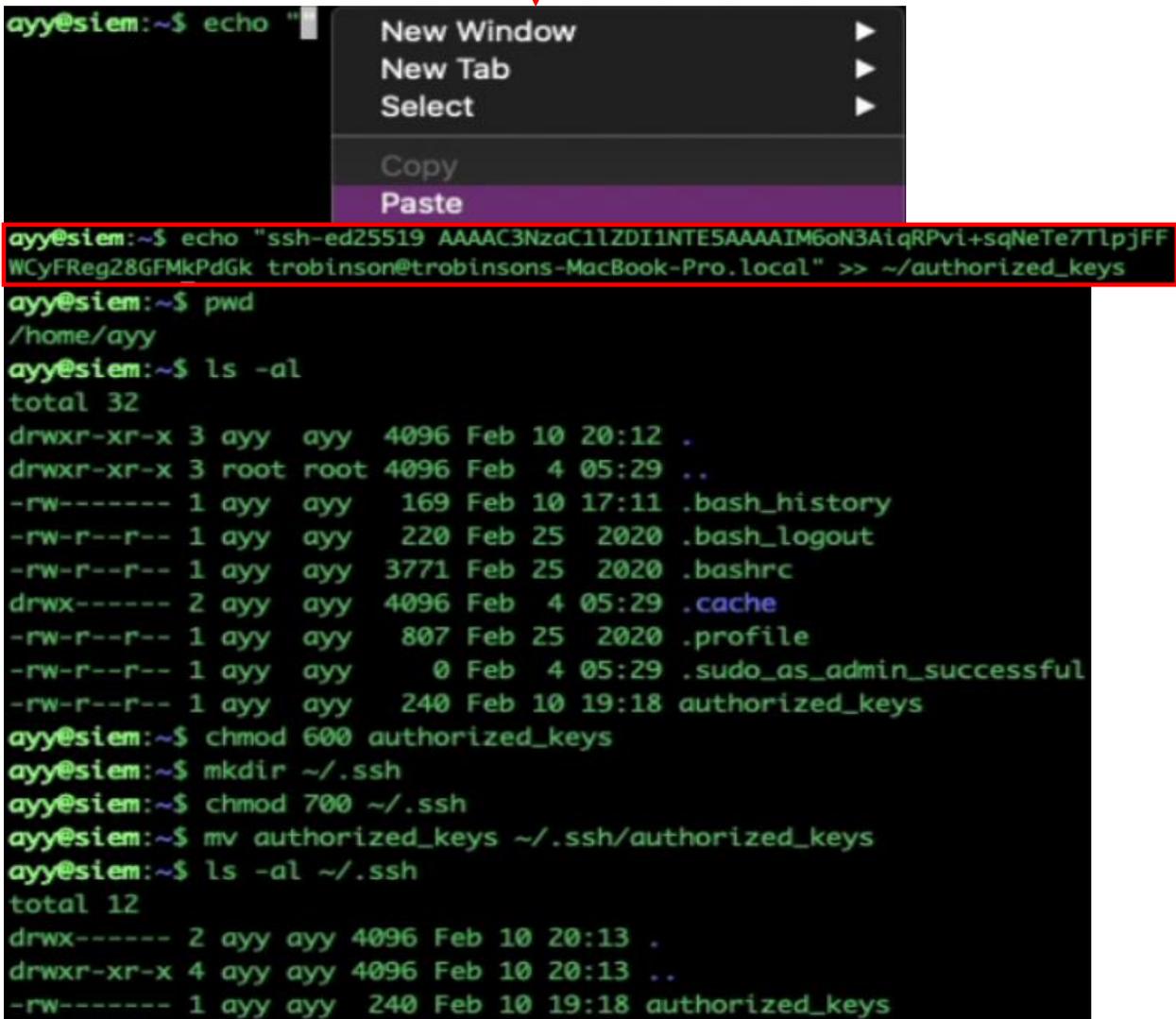
Continued to *fig. 15-42*

15-41: Open up two terminal sessions on the hypervisor host. In one session, open up an SSH session to the target virtual machine. Afterwards, in the second terminal session on the local system, run the command:

```
cat ~/.ssh/id_ed25519.pub
```

Highlight the content of the public key file that appears, then right click and select *Copy* from your terminal application's right-click context menu.

Continued from *fig. 15-41*



```
ayy@siem:~$ echo "
ayy@siem:~$ pwd
/home/ayy
ayy@siem:~$ ls -al
total 32
drwxr-xr-x 3 ayy ayy 4096 Feb 10 20:12 .
drwxr-xr-x 3 root root 4096 Feb  4 05:29 ..
-rw----- 1 ayy ayy 169 Feb 10 17:11 .bash_history
-rw-r--r-- 1 ayy ayy 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 ayy ayy 3771 Feb 25 2020 .bashrc
drwx----- 2 ayy ayy 4096 Feb  4 05:29 .cache
-rw-r--r-- 1 ayy ayy 807 Feb 25 2020 .profile
-rw-r--r-- 1 ayy ayy  0 Feb  4 05:29 .sudo_as_admin_successful
-rw-r--r-- 1 ayy ayy 240 Feb 10 19:18 authorized_keys
ayy@siem:~$ chmod 600 authorized_keys
ayy@siem:~$ mkdir ~/.ssh
ayy@siem:~$ chmod 700 ~/.ssh
ayy@siem:~$ mv authorized_keys ~/.ssh/authorized_keys
ayy@siem:~$ ls -al ~/.ssh
total 12
drwx----- 2 ayy ayy 4096 Feb 10 20:13 .
drwxr-xr-x 4 ayy ayy 4096 Feb 10 20:13 ..
-rw----- 1 ayy ayy 240 Feb 10 19:18 authorized_keys
```

15-42: Switch back to the terminal window with the active SSH session, and run:

```
echo "[right click -> paste]" >> ~/authorized_keys
```

To dump the contents from the hypervisor host's clipboard (e.g., the public key file) to the echo command. The echo command redirects the output to the `authorized_keys` file on the target lab VM, in that user's home directory. From here, students utilize the same commands from method 2 to place the `authorized_keys` file in the correct location, with the correct file and folder permissions. Refer to *fig. 15-40* for the complete list of commands. This series of illustrations demonstrates the process on the SIEM VM. Students will need to repeat this process on the IPS and Kali virtual machines as well.

15.3.3.3 Testing Key-Based Authentication

At this point, students should have generated a public/private key pair to enable key-based authentication. The public key, `id_ed25519.pub` should have been transferred to the SIEM, IPS and Kali VMs, renamed `authorized_keys`, and placed into the `~/.ssh` directory with specific file and folder permissions that allow only the user who owns the directory to access. The next step just verifying key-based authentication works as intended. Fortunately, its incredibly simple to do this. Establish an SSH session to the SIEM, IPS, and Kali virtual machines:

```
ssh siem
ssh ips
ssh kali
```

Note: If you went the alternative route and decided to used aliases just run:

```
siem
ips
kali
```

Depending on whether or not students password protected their private key file (`id_ed25519`), and key-based authentication is configured properly, establishing an SSH session will either require a password to decrypt the private key (two-factor authentication) or will be instantaneous.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
Enter passphrase for key '/Users/trobinson/.ssh/id_ed25519':
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
```

15-43: In order to test key-based authentication to lab virtual machines, students should SSH to their lab VMs. If they get one of the responses in the illustrations above (e.g., instantaneous login, or a prompt to enter the password for their private key), then configurating key-based authentication was successful. If not, check out [section 15.4, Troubleshooting SSH Connectivity and Key-Based Authentication](#), pp. 809-812. Be sure to confirm key-based authentication is working for the SIEM, IPS and Kali VMs.

15.4 Troubleshooting SSH Connectivity and Key-Based Authentication

If students are here, it means that there are problems either with SSH connectivity in general, or with key-based authentication. This section will cover some troubleshooting steps to follow, and settings to check to help find the root cause of various problems related to SSH connectivity.

- **Is the power on?**

If you're having problems establishing an SSH session to the lab virtual machines, start with the basics: Are the virtual machines powered on? I know it sounds ridiculous, but make sure the VMs are powered on before doing more complex troubleshooting tasks. All the VMs need to be powered on in order to access them, but remember that there are static routes that use the pfSense VM in order to reach the IPS network segment.

- **Is the SSH service enable and running?**

If students followed the installation instructions for the SIEM and IPS VMs, the SSH service should be installed and running by default on the Ubuntu Server VMs. kali VM is unique because it has the SSH service installed, but its disabled and not running by default. Instructions on how to enable and start the service were provided in [section 15.1.4](#) (p. 742).

- **Is the host-only network interface configured correctly? (correct IP, subnet mask, static route, etc.)**

Make sure that the host-only network interface for your host has the correct IP address and subnet mask configured to access the management network segment virtual machines (e.g., SIEM, and IPS), and that it has a static route to the IPS network segment in order to route network traffic to the kali VM. We covered configuring static routes for Windows, Linux, and macOS hypervisor hosts in sections [15.1.1](#), [15.1.2](#), and [15.1.3](#) (pp. 732-741).

- **Are the firewall rules on the LAN interface correct in order to allow access to the IPS/OPT1 network segment?**

Firewall rules and policies for the WAN, LAN, and OPT1 interfaces were covered in chapter 14. Students should check [section 14.4.4.2](#) (pp. 703-705) for a copy of the firewall rules students should have assigned to the LAN interface. Note that this step mainly applies to difficulties accessing the SSH service on the kali VM. However, if students add additional virtual machines to the IPS network they want to access over SSH, firewall rules need to be created to allow that access on the LAN interface.

- **Was the SSH key rejected?**

If after attempting to connect using key-based auth on Windows students get the error message:
server refused our key

Followed by a prompt to enter a password for the user, then something is wrong. Linux/MacOS users on the other hand, won't even get that prompt (by default – as a side project, look into enabling verbose mode for the ssh client) – they'll immediately be prompted to enter a password for the user. Below is a list of things to double check to help troubleshoot authentication issues:

- **What are the file permissions for ~/.ssh and the authorized_keys file?**

Students on Linux/MacOS hypervisor hosts who used ssh-copy-id can skip this step. ssh-copy-id handles file permissions on the user's behalf.

Most modern SSH servers are very insistent on the ~/.ssh directory and authorized_keys files having restrictive file permissions. This is to prevent malicious users from attempting to add their public keys to another user's ~/.ssh/authorized_keys file, and establish key-based authentication as them. As such, **make sure that the that the ~/.ssh directory itself has "700" permissions (chmod 700 ~/.ssh), and that the authorized keys file has "600" permissions.** The fastest way to check this is by running `ls -al ~/.ssh` on the remote system students are attempting to troubleshoot:

```
ayy@siem:~$ ls -al ~/.ssh
total 12
drwx----- 2 ayy ayy 4096 Feb 10 23:56 .
drwxr-xr-x  4 ayy ayy 4096 Feb 10 23:46 ..
-rw-----  1 ayy ayy  120 Feb 10 23:56 authorized_keys
```

15-44: output from `ls -al ~/.ssh`. Pay attention to the "." entry, and the authorized_keys entry.

In *fig. 15-44*, notice the "." and the authorized_keys entry. Note that the "." entry represents the file permissions for the ~/.ssh directory itself. The "." symbol is a special character in unix that always means "The current directory".

To the left of every file entry there are a bunch of symbols. The "." entry reads:

```
"drwx-----"
```

This is another way of representing file permissions for the ~/.ssh directory. The "d" symbol denotes that this file is a directory, while the rest of the line, "rwx-----" are another way of representing file permissions. In this case, these is the representation for "700" permissions. Meaning "only the owner of this directory can read, write, or execute (enter) this directory."

while authorized_keys reads:

```
"-rw-----"
```

This is another way of visualizing "only the owner of this file may read or write to it."

If the file permissions for ~/.ssh and the ~/.ssh/authorized_keys are different in any way, run:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

○ Is the authorized_keys file formatted correctly?

This is another troubleshooting step in which students who used ssh-copy-id can skip, because it handles formatting automatically. The authorized_keys file has a very specific format:

```
[type of key] [key content itself] [comment]
```

For example:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIM6oN3AiqRPvi+sqNeTe7T1pjFFWCyFReg28GFMkPdGk
trobinson@trobinsons-MacBook-Pro.local
```

ssh-ed25519 indicates this is the public key of key pair that is using the ed-25519 algorithm. The blob of letters and numbers in the center is the actual public key itself, and finally the text trobinson@trobinsons-MacBook-Pro.local is just a comment line, and is entirely optional. ssh-keygen on Linux/Unix/macOS is rational and the comment line tells users what user and remote system the public key entry in the authorized_keys file represents, whereas the comment line for puttygen is ed25519-key-20210107 – "I'm an ed25519 key, here is a timestamp for the date I was generated and no other relevant metadata".

All three portions represent a single authorized_keys entry. The authorized_keys file can contain more than one entry. **The two most important things to remember about the authorized keys file are that each entry in the authorized keys file is only a single line, and that the authorized keys file absolutely requires Unix line feeds (LF).**

That first point is extremely important. Every single `authorized_keys` entry is only a single line. Even if that line wraps around in your terminal session or text editor, they can only be a single line long. The fastest way to check this to establish an SSH session on the remote system who's `authorized_keys` file is malformed, and run the command `wc -l ~/.ssh/authorized_keys`:

```
ayy@siem:~$ wc -l ~/.ssh/authorized_keys
1 /home/ayy/.ssh/authorized_keys
```

15-45: output from the `wc -l` command.

The `wc` or word count command can be used to count the number of words in a file. with the `-l` option, it can be used to count the number of lines in a file. Notice the number "1" on the left portion of the illustration? That indicates that the `authorized_keys` file is exactly one line long. Given that the `authorized_keys` file entry for the lab virtual machines should only contain a single public key, we can rule out that the `authorized_keys` file has too many lines in it.

Note: Something peculiar I've noticed that seems to affect the Windows users copying their `authorized_keys` files to the lab VMs, is that if you run `wc -l`, the file will occasionally have zero lines in it, in spite of having a fully formed public key entry. This is because there's technically no line termination character in the file. Run the command:

```
echo "" >> ~/.ssh/authorized_keys
```

This will effectively tell the file to start a new, blank line.

```
ayy@ips:~$ wc -l ~/.ssh/authorized_keys
0 /home/ayy/.ssh/authorized_keys
ayy@ips:~$ echo "" >> ~/.ssh/authorized_keys
ayy@ips:~$ wc -l ~/.ssh/authorized_keys
1 /home/ayy/.ssh/authorized_keys
ayy@ips:~$ cat ~/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC11ZDI1NTE5AAAAIM2jwZn41UG7Iiajx2XSEqdYWx01QsN5Zq7Sv6Ih+Laf ed25519-key-20210107
```

15-46: Sometimes, for reasons beyond me, the public key entry will be written to the `authorized_keys` file without an end of line. This will cause `wc -l` to say "yeah, this file has zero lines in it." This can be fixed with the command:

```
echo "" >> ~/.ssh/authorized_keys
```

This command essentially tells the file "Append a blank line to this file, please."

Now, the other major concern I mentioned is verifying that the EOL conversion to Unix (LF) was performed correctly. For the most part, this is mainly a concern for the Windows users, but it doesn't hurt to double check it. Check out [section 15.2.3.1, pp. 750-759](#).

15.5 (Optional Content) Remote Access Enhancements

This section contains a couple of optional features that, I didn't really feel were "core" to this chapter, but are configuration options that will either allow students to enhance the ease of access to the lab environment, enhance the security of their lab environment, or even strike a balance between accessibility and security. In this section, we will cover enabling SSH authentication as the root user, and disabling password authentication for SSH.

15.5.1 Enabling SSH Access as the root User

There are a number of information security-minded individuals who believe that nobody should be able to log in to a system using the root account, let alone allowing remote access as the root user. In fact, the security model for Ubuntu Linux is based off users not being able to directly log in as the root user – Ubuntu users are expected to utilize the `sudo` command to execute other commands or applications that require root privileges. Out in the real-world use of the root account is heavily restricted as well because the slightest mistake running a command or applying a configuration setting can trigger catastrophic results in the form of outages or worse. It's easy for humans to make mistakes, and what's more Linux/Unix-based systems don't come with guardrails to protect users from themselves. This is why root access is heavily restricted, and when used, should be heavily audited in order to know who did what as the root user and when.

When it comes to enabling SSH access as the root user in the real-world, it's usually an extremely bad idea, at least not without tons of auditing and protection in place to ensure that it isn't abused by unauthorized users. However, we've built a lab environment together – your lab environment. If ever there was a place to make mistakes and learn the risks and consequences of certain actions, this is probably it. Enabling SSH access as the root user is a two-part process. First, students need to confirm that the `PermitRootLogin` directive in `sshd_config` will actually allow the root user to log in. Then, we need to enable key-based authentication as the root user. Let's get started.

The SSH service relies on a configuration file, `/etc/ssh/sshd_config`. This file is responsible for setting a wide variety of configuration parameters. As mentioned above, the configuration option we're interested in is the `PermitRootLogin` directive. This option can be set to `yes`, `no`, `without-password`, and `prohibit-password`. The `yes` and `no` options are pretty self-explanatory – either the service will allow the root user to log in with no restrictions (`yes`), or completely forbids it (`no`). The `without-password` and `prohibit-password` options operate in an identical manner – the root user can log in via SSH, but they cannot log in using a password (technically, the `without-password` option is considered deprecated. This means that it should no longer be used because the developers may consider it an invalid configuration option in a future release). This configuration strikes a balance between security and accessibility. It allows users to configure remote login as the root user if they want, but protects the account against brute-force attacks. A brute-force

attack is when an adversary attempts to repeatedly guess the password an account, in the hopes that they're able to successfully guess the password, and gain access.

Most Linux distributions that include the OpenSSH service as a software package default to setting the `PermitRootLogin` option to `prohibit-password`. If students want to check this on their own, here is a simple command that can be used to check the `PermitRootLogin` setting in the `/etc/ssh/sshd_config` file:

```
grep PermitRootLogin /etc/ssh/sshd_config
```

The `grep` command allows users to search for a pattern in a file, or group of files. In this case, we're looking for the pattern `PermitRootLogin`, in the file `/etc/ssh/sshd_config`. In both Kali Linux, and Ubuntu Server 20.04, the default for this setting will be:

```
#PermitRootLogin prohibit-password
```

Most Linux/Unix configuration files use the `"#"` to denote that everything else in this line should be ignored, or treated as a comment. The `sshd_config` file does this, but has a special notice at the beginning of the configuration file:

```
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
```

What this means is that even though the `PermitRootLogin` line we found is commented out, that `prohibit-password` is considered the default setting. If a user wanted to override this default, they would remove the `"#"` at the beginning of the line, and change the `PermitRootLogin` setting as they desire.

So now that students have established that they can log in as the `root` user so long as they are not using password authentication, the next step is enabling key-based authentication for the `root` user. Our usual method of doing this involves copying the public key of an SSH keypair using the copy methods we went over in this chapter. However, at this point if students have successfully established key-based authentication as their normal account on the SIEM, IPS and Kali virtual machines, they can copy the `~/ .ssh` directory of that user to the `root` user's home directory and change the file ownership to the `root` user. I will demonstrate the process on the SIEM virtual machine, and students can repeat the process on the IPS and Kali VMs on their own. Establish an SSH session on the SIEM VM and run the following commands:

```
sudo su -
cp -r /home/[username]/.ssh/ ~/
chown -R root:root ~/.ssh
```

The first command should look familiar. Students will utilize `sudo su -` to become the `root` user. From there, we use the `cp` command with the `-r` option to (r)ecursively copy everything in the user's `.ssh` directory to the `root` user's home directory. Be sure to replace `[username]` with the

actual username of your user account. For example, to recursively copy the `.ssh` directory of the `ayy` user:

```
cp -r /home/ayy/.ssh ~/
```

Finally, the `chown` command allows students to (ch)ange the (own)ership of the files they just copied to root's home directory. The `-R` option tells the command to change the file ownership not just for the `~/ .ssh` directory, for the files contained within (e.g., the `authorized_keys` file). Remember that the `~/ .ssh` directory and the files contained therein must have restrictive permissions to where only the individual user can access them. Fortunately, since students already handled file permissions earlier, the only thing that needs to be modified is the owner of the file. Once these tasks are completed, students may exit their session as the root user, close the SSH session, then repeat this process on the IPS and Kali VMs, if desired.

```
ayy@siem:~$ grep PermitRootLogin /etc/ssh/sshd_config
#PermitRootLogin prohibit-password
# the setting of "PermitRootLogin without-password".
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# cp -r /home/ayy/.ssh ~/
root@siem:~# chown -R root:root ~/.ssh
root@siem:~# exit
logout
ayy@siem:~$ exit
```

15-47: In order to enable SSH login as the root user, students need to confirm that its even allowed by SSHD in the first place. First, check the `PermitRootLogin` directive by using the `grep` command to search for that string in `/etc/ssh/sshd_config`. In the illustration above, it is confirmed that the default setting is `prohibit-password`, meaning that in order to log in as root, students will need to establish key-based authentication for the root user.

Fortunately, students should have already configured key-based auth for the standard user account already, so its just a matter of copying the standard users `.ssh` directory to the root user's home directory, then changing the file ownership so that root owns the `.ssh` directory and all of the files contained within – specifically, the `authorized_keys` file. After completing this process on the SIEM VM, students may repeat these steps on the IPS and Kali virtual machines, if desired.

15.5.1.1 Testing root SSH for Linux/macOS Hypervisor Hosts

Once logged out, students can test their access. For the Linux/macOS users, run:

```
ssh root@siem
```

The ssh command should be able to read the SSH config file and connect to the IP address of the SIEM VM, while the root@ section allows students to override the User setting in the config file with the root user. If students did not set up an SSH config file, run:

```
ssh root@172.16.1.3
```

Of course, replace 172.16.1.3 with the IP address of the SIEM VM as necessary. If everything is working correctly either students will be prompted for the password to their SSH private key, or will be instantaneously logged in as root.

```
trobenson@trobinsons-MacBook-Pro ~ % ssh root@siem
Enter passphrase for key '/Users/trobenson/.ssh/id_ed25519':
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)

root@siem:~# whoami
root
```

15-48: Run `ssh root@siem` to test whether or not key-based authentication as the root user was configured properly. Depending on whether students opted to password protect their private keys, they will either be prompted for the password to decrypt the private key, or instantly provided with an SSH session as the root user, if everything is working properly.

Students can create multiple unique SSH config connection profiles for the same IP address, but connecting as different users. Here is an example config file you can use, using the IP addresses and usernames from my lab environment:

<https://gist.github.com/da667/24541166284f84dfc4eb314b4aa4246b>

For students lacking in internet access, modify the existing `~/.ssh/config` file as necessary, using the template below:

SSH config file with root user connection profiles

```
Host siem
    Hostname 172.16.1.3
    User ayy

Host siemroot
    Hostname 172.16.1.3
    User root

Host ips
    HostName 172.16.1.4
    User ayy

Host ipsroot
    HostName 172.16.1.4
    User root

Host kali
    HostName 172.16.2.2
    User ayy

Host kaliroot
    HostName 172.16.2.2
    User root
```

As always, substitute IP addresses in the Hostname field, and usernames in the User field as necessary. Using this config file, students could SSH to the SIEM VM as the ayy user by running:

```
ssh siem
```

Or SSH to the SIEM VM as the root user by running

```
ssh siemroot
```

The same applies for the ipsroot, and kaliroot connection profiles – so long as key-based authentication has been configured for the root on the IPS and Kali virtual machines, these Host profiles may be used to connect to the IPS and Kali VMs over SSH as the root user.

```
trobenson@trobinsons-MacBook-Pro ~ % cat ~/.ssh/config
Host siem
    Hostname 172.16.1.3
    User ayy

Host siemroot
    Hostname 172.16.1.3
    User root

Host ips
    HostName 172.16.1.4
    User ayy

Host ipsroot
    HostName 172.16.1.4
    User root

Host kali
    HostName 172.16.2.2
    User ayy

Host kaliroot
    HostName 172.16.2.2
    User root
```



```
trobenson@trobinsons-MacBook-Pro ~ % ssh siemroot
Enter passphrase for key '/Users/trobenson/.ssh/id_ed25519':
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)

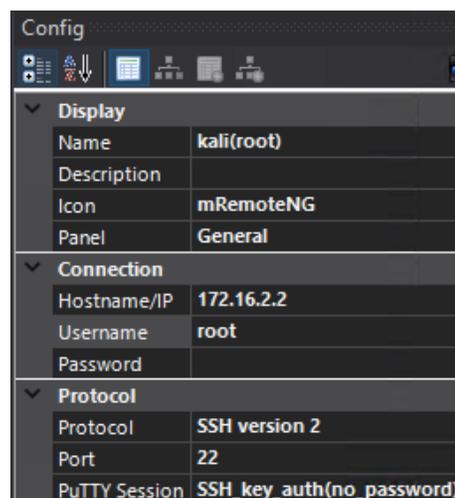
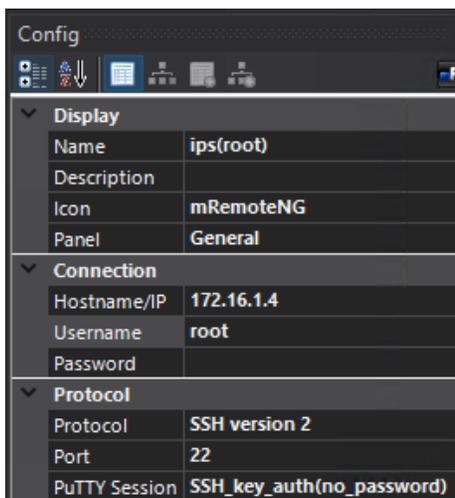
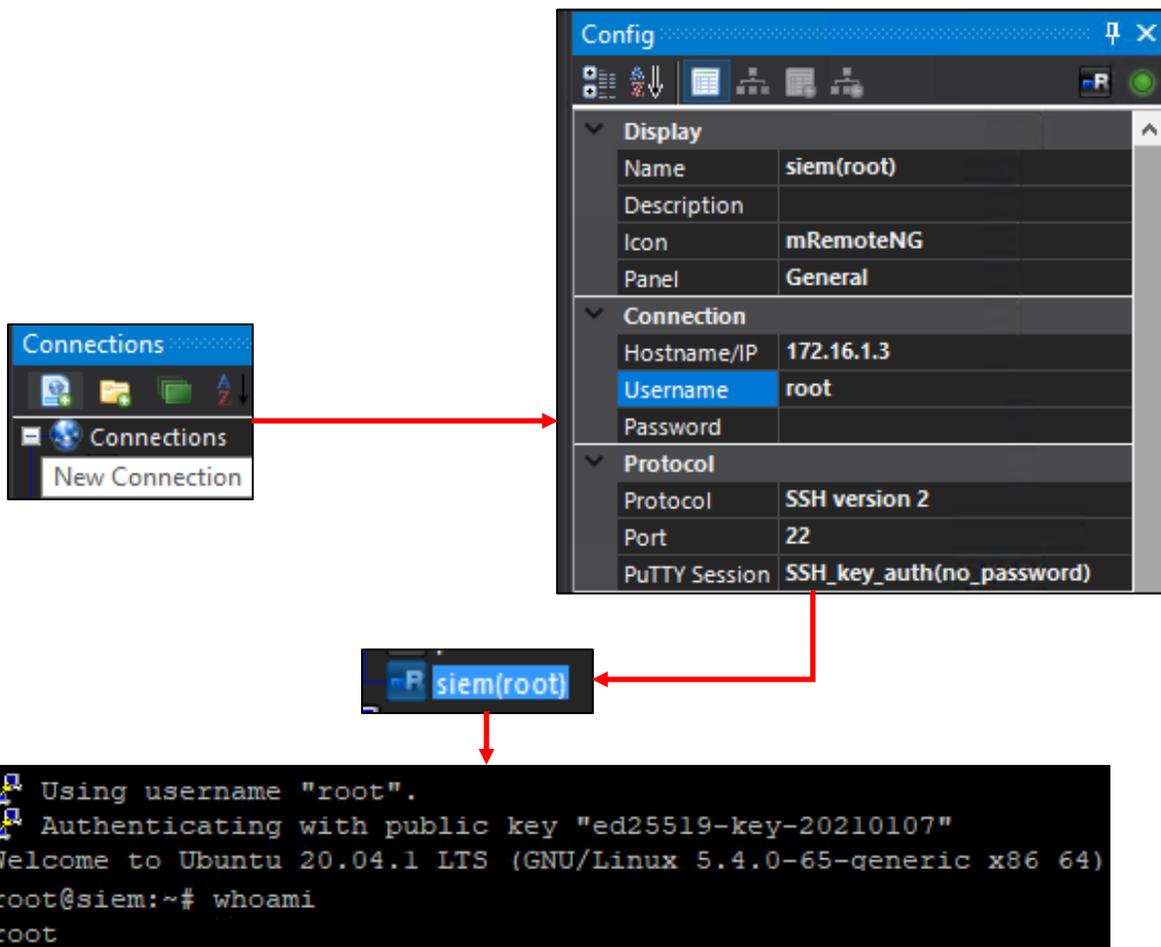
root@siem:~# whoami
root
```

15-49: If students wish to further enhance their laziness, they can create specific Host profiles in their SSH config file that will allow them to connect to the SIEM, IPS and/or Kali virtual machines as the root user, so long as key-based authentication has been properly enabled on each of those virtual machines. This enables students to run `ssh siemroot` and log in as the root user via key-based auth automatically. Recall that the SSH config file also enables shortcuts for the `scp` command as well, meaning students can copy files to and from their lab virtual machine as the root user if necessary.

15.5.1.2 Testing root SSH for Windows Hypervisor Hosts

Windows users on the other hand, will need to create a new mRemoteNG connection profile in order to test this functionality. The process for this should be second nature by now, but just to review:

- Click the *New Connection* icon in the *Connections* pane. Name the new connection profile something like `siem(root)` to indicate its purpose
- In the *Config* pane for the new connection profile, under the *Protocol* section, click the *Protocol* drop-down, and select *SSH version 2*. In the *PuTTY Session* field drop-down, select the putty session we created together earlier in order to enable key-based authentication. Under the Connection portion, enter the IP address of the SIEM VM in the *Hostname/IP* field, and enter `root` in the *Username* field. As always, leave the *Password* field blank
- Double-click `siem(root)` connection profile under the *Connections* pane. If everything is working correctly students will either receive a prompt to enter the password to decrypt the SSH private key, or an instantaneous SSH connection to the SIEM VM as the root user
- If successful, students may create additional mRemoteNG connection profiles for the IPS and Kali virtual machines as well, then to confirm successful login as the root user over SSH



15-50: Windows students on the other hand will need to create a new mRemoteNG connection profile to test whether or not root access over SSH is working. Create a new profile, and in the config section, remember to set the Protocol to SSH version 2, and the PuTTY Session field to the custom putty session we created together in this chapter for enabling key-based authentication. From there, enter the IP address of the SIEM VM into the Hostname/IP field, and the Username root, then test the connection profile to see if it yields an SSH session as the root user. If successful, repeat the process for the IPS and Kali virtual machines, if desired.

15.5.1.3 Remember, This isn't Strictly Necessary

In future chapters, certain software will require root access to be installed or run properly. However, if you are not comfortable with enabling SSH access for the root user that it is not necessarily. Remember that `sudo su` - is a perfectly viable alternative to elevate to the root account as needed. However, **regardless of whether or not students intend to configure SSH access as the root user, I would highly advise enabling the security enhancements in section 15.5.2 to disable password authentication over SSH entirely for all users.**

15.5.2 Disabling password authentication over SSH

While the root account is relatively safe, with that `PermitRootLogin prohibit-password` directive, standard user accounts are not protected, and opportunistic attackers could still attempt to brute force the user accounts. Once an attack has access as a standard user, they can seek to elevate their privileges, perhaps through a misconfiguration, or a privilege escalation vulnerability. Remember that `sudo su` - only requires the standard user's password to elevate to the root account.

While this is a lab environment and the risk of anybody attempting to attack a student's lab virtual machines to access them is, admittedly, not a huge risk, knowing how to secure SSH access is extremely valuable, so I think the benefits of learning how to configure this enhanced security are worth teaching. Students should be aware that if they enable these security enhancements then somehow misplace or lose their SSH private keys (or the password to password-protected private keys) that they will effectively lock themselves out of SSH access to their SIEM, IPS, and Kali virtual machines.

In such an event, students may opt to make use of snapshots or save states to revert their virtual machines to start prior to making these changes. However, this could result in the loss of configuration data, or information so in addition to maintaining regular snapshots, **It is highly recommended that students make a backup of the `/etc/ssh/sshd_config` file on the SIEM, IPS and Kali virtual machines prior to making these changes,** that way, if users lose their SSH key pairs, the default SSHD configuration can be restored (by logging in to the affected virtual machines using the hypervisor's virtual console), and students can reconfigure key-based authentication using a newly generated SSH key pair.

15.5.2.1 Backing Up (and Restoring) the `/etc/ssh/sshd_config` file

To make a backup of the `sshd_config` file, run the following command:

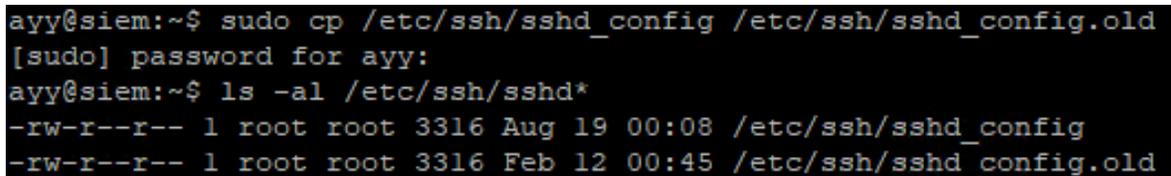
```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
```

The `/etc/ssh/sshd_config` file requires root access to modify it, so `sudo` is used to perform the `cp` command against `/etc/ssh/sshd_config`, and make a copy named `sshd_config.old` in the `/etc/ssh` directory.

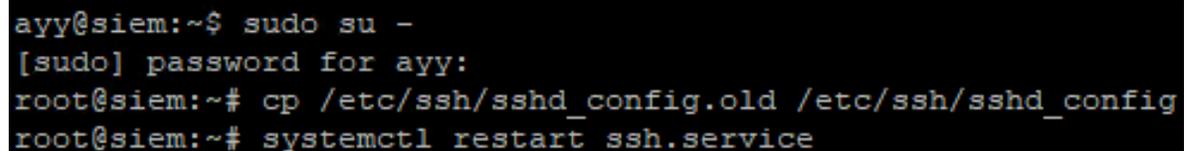
If for some reason students need to restore the `sshd_config` back to its default state, run the following commands:

```
sudo su -
cp /etc/ssh/sshd_config.old /etc/ssh/sshd_config
systemctl restart sshd.service
exit
```

In this case, we're using `sudo su -` to become root. We use the `cp` command to make a copy of the `sshd_config.old` file and overwrite the existing `sshd_config`, effectively restoring the default configuration. However, the SSH service will not honor the new configuration unless the service has been restarted, so that is why students must run `systemctl restart ssh.service` afterwards as the root user. Finally, students exit their session as the root user.



```
ayy@siem:~$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
[sudo] password for ayy:
ayy@siem:~$ ls -al /etc/ssh/sshd*
-rw-r--r-- 1 root root 3316 Aug 19 00:08 /etc/ssh/sshd_config
-rw-r--r-- 1 root root 3316 Feb 12 00:45 /etc/ssh/sshd_config.old
```



```
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# cp /etc/ssh/sshd_config.old /etc/ssh/sshd_config
root@siem:~# systemctl restart ssh.service
```

15-51: Since students are about to make significant changes to `/etc/ssh/sshd_config`, it is recommended to make a backup of this file on the SIEM, IPS, and Kali VMs before modifying it. The first screen capture above is a single command that allows students back up the `sshd_config` to `sshd_config.old` prior to making any modifications. The second image illustrates the process of restoring the default `ssh_config` from the `sshd_config.old` file, then restarting the SSH service so that it reads and applies the "new" configuration. This process can be done in the event that users lock themselves out of SSH access in the future, and want to restore access without reverting an old snapshot.

15.5.2.2 Modifying the PasswordAuthentication, ChallengeResponseAuthentication, and AuthenticationMethods directives

With a backup of the `sshd_config` file in place, the next step comes in making the necessary changes to disable password authentication. Specifically, students will need to modify the `PasswordAuthentication`, `ChallengeResponseAuthentication`, and `AuthenticationMethods` configuration options, to completely disable password authentication over SSH. Run the following command on the SIEM, IPS, and Kali virtual machines:

```
egrep "ChallengeResponseAuthentication|AuthenticationMethods|PasswordAuthentication" /etc/ssh/sshd_config
```

The `egrep` command is a special version of `grep` that allows the use of regular expressions. Without getting too deep in the weeds, this `egrep` command says "Show me any lines in the `/etc/ssh/sshd_config` file that contain the strings `ChallengeResponseAuthentication` OR `AuthenticationMethods` OR `PasswordAuthentication`." The output on the SIEM and IPS virtual machines, running Ubuntu Server 20.04 should look something like this:

```
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
PasswordAuthentication yes
```

15-52: These are the results from the `egrep` command that students should see for any VM running Ubuntu Server. There should be no results for `AuthenticationMethods`.

The highlighted portions of *fig. 15-52* contain some of configuration directives we're searching for. From this output, we can determine that `ChallengeResponseAuthentication` is already set to `no`, `PasswordAuthentication` is set to `yes`, and the `AuthenticationMethods` directive has not been set yet. Be aware that there will be no results for `AuthenticationMethods` in a default `sshd_config` file. Let's change that by running the following commands:

```
sudo su -
sed -i 's#^PasswordAuthentication yes#PasswordAuthentication no#'
/etc/ssh/sshd_config
echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only."
>> /etc/ssh/sshd_config
echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
egrep "ChallengeResponseAuthentication|AuthenticationMethods|PasswordAuthentication"
/etc/ssh/sshd_config
systemctl restart ssh.service
exit
```

This collection of commands starts with students using `sudo su -` to gain root access. Next, students use the `sed` command to modify a specific line in the `sshd_config` file. The `sed` command says "Find the first line in the `/etc/ssh/sshd_config` file that begins with `PasswordAuthentication yes`, and change that to `PasswordAuthentication no`." Next, students use the `echo` command with output redirection to add comment line, and the `AuthenticationMethods` directive set to `publickey`. This setting forces the SSH server to only accept key-based authentication. Finally, students restart the SSH service, then exit their session as the root user. **Students may perform these commands on both the SIEM and IPS virtual machines to effectively disable password authentication over SSH.**

```
~# sed -i 's#^PasswordAuthentication yes#PasswordAuthentication no#' /etc/ssh/sshd_config
~# echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config
~# echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
PasswordAuthentication no
#Adding AuthenticationMethods publickey to force key-based auth over SSH only.
AuthenticationMethods publickey
~# systemctl restart ssh.service
```

15-53: As the root user, use the `sed` command to change `PasswordAuthentication` from `yes` to `no`, and the `echo` command with output redirection to add the `AuthenticationMethods publickey` parameter. Next, `egrep` is used to confirm that the `sed` and `echo` commands modified the configuration file correctly. Finally, students restart the SSH service so that the service will read and implement the changes in the `sshd_config` file. Remember, these specific `sed` commands should only be ran on the SIEM, and IPS VMs (or additional Ubuntu Server VMs students add to their lab environment).

The Kali VM however, has a slightly different default configuration for its `/etc/ssh/sshd_config` file:

```
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
```

15-54: Notice that the `PasswordAuthentication` directive includes a `"#"` symbol directly in front of it. This `sshd_config` is functionally identical to the `sshd_config` on our Ubuntu systems (SIEM and IPS). However, the `sed` command used to edit that line for the Kali VM will need to be modified slightly. Once again, `ChallengeResponseAuthentication` should default to `no`, and `AuthenticationMethods` should not be set.

The highlighted portion of fig. 15-54 is the only difference we care about in the `sshd_config` file on the Kali VM, compared the same file on the IPS and SIEM virtual machines. However, because it is ever so slightly different, students will need to run a slightly different `sed` command to modify that line. **Run the following commands on the Kali VM:**

```
sudo su -
sed -i 's/^#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >>
/etc/ssh/sshd_config
echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
egrep "ChallengeResponseAuthentication|PasswordAuthentication|AuthenticationMethods"
/etc/ssh/sshd_config
systemctl restart ssh.service
```

Again, the only command that is slightly different looking is the `sed` command used to disable the `PasswordAuthentication` directive. This command tells `sed`, "Find the first line in the `/etc/ssh/sshd_config` file that contains `#PasswordAuthentication yes` and change that to `PasswordAuthentication no`". Otherwise, the remaining commands are exactly the same, and the end result is a Kali VM that no longer accepts password authentication over SSH.

```
~# sed -i 's/^#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
~# echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config
~# echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
PasswordAuthentication no
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
#Adding AuthenticationMethods publickey to force key-based auth over SSH only.
AuthenticationMethods publickey
~# systemctl restart ssh.service
```

15-55: The slight change in the Kali Linux `sshd_config` file means we have to use a slightly different `sed` command to change the `PasswordAuthentication` directive to `no` on the Kali VM. Other than that, the remaining commands are identical to the ones ran on the SIEM and IPS virtual machines and/or Ubuntu server bastion hosts.

You sed it, brother

Please note that students do not have to use the `egrep` and/or `sed` commands to view or modify the necessary lines in the `sshd_config` file on the bastion host, SIEM, IPS, or Kali systems. If you're comfortable reading files using other command line utilities such as `less`, `more`, `cat`, etc, you're more than welcome to use any other utilities you are more comfortable with. Likewise, if you'd rather modify the necessary lines in the `sshd_config` file on the lab virtual machines using other text editors, students are welcome to use `vi`, `nano`, `ed`, `emacs` or any other text editor they are comfortable with to modify the configuration files as necessary.

If you're open to suggestion, it's worth learning how to edit configuration files using `vi`, mainly because every Linux/Unix-based system has it by default. Not to mention, these configuration directives may change again in the future. What better place to learn how to use a command-line text editor than in your lab environment?

No matter what method or utilities you use to modify the the `sshd_config` file, make absolutely sure that you made a backup of the original file before you begin. That way if there is a problem, there is a known good backup to restore from.

15.5.2.3 Verifying Password Authentication over SSH is disabled

Now students will need to verify that password authentication over SSH is disabled. The process for doing this on Windows and Linux/macOS is slightly different. In this section I will demonstrate how to confirm password authentication is disabled for the SIEM VM for Windows users as well as Linux/macOS users.

Windows Users:

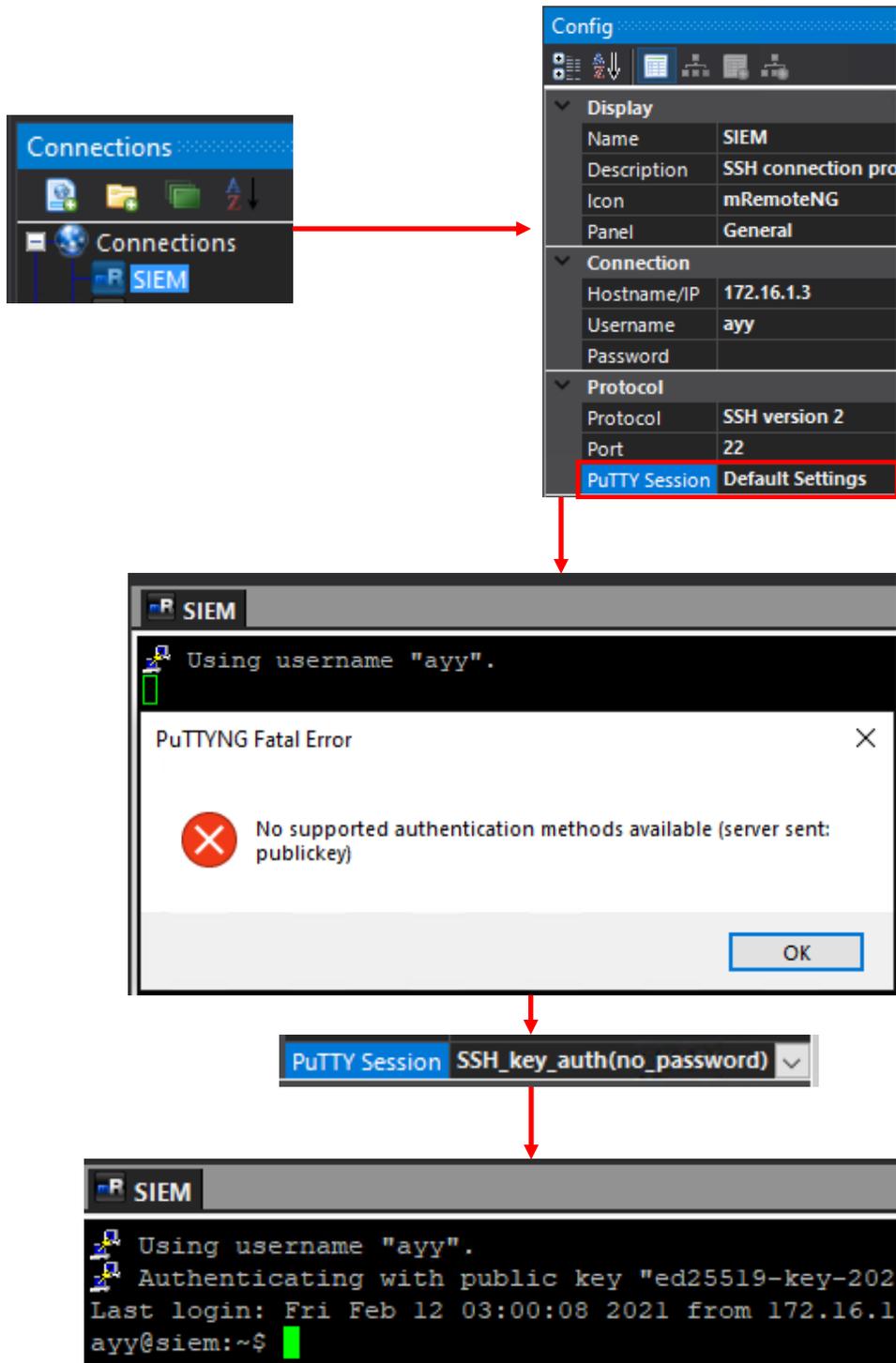
Open mRemoteNG. In the *Connections* pane, click on SIEM connection profile to highlight it. In the *Config* pane, navigate to the *PuTTY Session* field under the *Protocol* section, and select *Default Settings* from the drop-down menu. Next, double click on the SIEM connection profile icon in the *Connections* pane to open a session to the SIEM VM.

Upon attempting to connect to the SIEM VM over SSH, students will immediately be greeted with a window titled *PuTTYNG Fatal Error*:

```
No supported authentication methods available (server sent: publickey)
```

Congratulations! This window is more or less telling the user that password authentication is not available. Click the *OK* button to close the window, and change the PuTTY Session back to the session created earlier in this chapter to enable key-based authentication. Afterwards try connecting again. Repeat this process on the IPS and Kali virtual machines to ensure password authentication over SSH is disabled on those systems as well.

Note: We covered this in the sidebar conversation in Section 15.2.3.3. However, it is important enough that this bears repeating: **If you decided to disable password authentication over SSH on your lab virtual machines, WinSCP will no longer work without configuring key-based authentication.** If you followed the instructions for disabling password authentication via SSH to the lab virtual machines, make absolutely sure you read the sidebar conversation, **Bonus Lesson: Key-Based Authentication with WinSCP** (pp. 782-783), and **enable key-based auth for your WinSCP site profiles, if you need to transfer files to your lab virtual machines later.**



15-56: The method for testing if password authentication over SSH is disabled for Windows clients is very simple. Modify the SIEM connection profile's *PuTTY Session* setting back to *Default Settings*, then attempt to connect to the SIEM VM over SSH. If students get an error notification stating there are no supported authentication methods available, everything is working as intended. Students can then change the *PuTTY Session* setting back to the session created earlier to enable key-based authentication, and verify SSH access is working properly once more. Repeat this process for the IPS and Kali virtual machines.

Linux/MacOS Users:

If the process for testing access on Windows was simple, then this is dead simple. Run the following commands. Please note that this demonstration assumes students set up an SSH config file with a Host profile for the SIEM VM.

```
mv ~/.ssh/id_ed25519 ~/.ssh/id_ed25519.old
ssh siem
```

If everything is working properly, students will get a single line error message:

```
ayy@172.16.1.3: Permission denied (publickey).
```

If this error message appears then that means the configuration changes were successful! Next, run the following commands:

```
mv ~/.ssh/id_ed25519.old ~/.ssh/id_ed25519
ssh siem
```

Students should be able to once more successfully establish an SSH session to the SIEM VM. Repeat this process on the IPS and Kali virtual machines to ensure password authentication over SSH is disabled on those systems as well.

Note: I'm going to give you Linux/MacOS users mostly the same conversation I gave the Windows users above: the `scp` command, or an SCP/SFTP client is going to be the best and most secure method of copying files to or from your lab environment. **Ensure whatever SCP/SFTP client you're using has a way for you to specify SSH private keys to use for authentication, if you choose to disable password authentication over SSH.** Fortunately, the `scp` command knows to do key-based authentication by default, and knows how and where to find the SSH private keys (`~/.ssh`) without any further setup.

```
trobinsont@trobinsonts-MacBook-Pro ~ % mv ~/.ssh/id_ed25519 ~/.ssh/id_ed25519.old
trobinsont@trobinsonts-MacBook-Pro ~ % ssh siem
ayy@172.16.1.3: Permission denied (publickey).
trobinsont@trobinsonts-MacBook-Pro ~ % mv ~/.ssh/id_ed25519.old ~/.ssh/id_ed25519
trobinsont@trobinsonts-MacBook-Pro ~ % ssh siem
Enter passphrase for key '/Users/trobinsont/.ssh/id_ed25519':
Last login: Thu Feb 11 18:40:12 2021 from 172.16.1.2
ayy@siem:~$
```

15-57: In order to test whether or not password authentication over SSH is disabled on the SIEM VM, use the `mv` command to rename the `id_ed25519` file (the private key) to literally anything else. In this case, `id_ed25519.old`. Afterwards, run `ssh siem`. Students should immediately get an error message `Permission denied (publickey)`. Use the `mv` command to rename the `id_ed25519.old` back to `id_ed25519`. Once more, run the command `ssh siem` and verify the ability to successfully connect to the SIEM VM over SSH, using key-based authentication. Repeat this process for the IPS and SIEM virtual machines.

Laziness, or Security?

There are a lot of die-hard cybersecurity professionals out there who will tell you that every single SSH key you generate and use should be password protected. Well, I don't necessarily agree with that. I believe in presenting you with choices, and demonstrating the trade-offs.

This is a lab environment that is meant to be easy to stand up and modify at a moment's notice. If it's annoying, or otherwise difficult to access your VMs, you're not going to want to use the environment, right? If you set up key-based authentication without password protecting your private key, it's extremely fast and convenient to access your virtual machines remotely. Fast and easy remote access to your systems so you can focus on whatever projects you're working on!

But if that's the case, why am bothering showing you how to password protect your SSH key and set two-factor authentication at all? Because, in the real world, it's considered the right way™ to do key-based authentication. I'm showing you how to do things the right way™ so that when you get out there in a real world, you're aware of how to set it up quickly and safely. What better place to make sure you got the basics down than in your lab environment?

Bear in mind, we didn't cover some of the more arcane and maddening authentication methods that SSH can support (for instance, TOTP challenge/response, push authentication, and Yubikey/FIDO authentication), but I'll leave those as things you can research and look in to yourself.

15.6 Chapter Review

Congratulations on making it to the end of another chapter! The purpose of this chapter was to set students running hosted hypervisors on Windows (e.g., VirtualBox, VMware Workstation, Microsoft Client Hyper-V) or Linux/MacOS (VirtualBox, VMware Workstation, VMware Fusion) up for success for the remaining lab configuration chapters. By the end of this chapter, students will have gained a better understanding of:

- Network routing, why it matters, and what role static routing plays in ensuring students can access all of the resources in their lab environment
- Utilizing the Secure Shell (SSH) protocol to enable secure and convenient remote access to their lab virtual machines. Students on Windows got familiar with mRemoteNG and setting up connection profiles, while Linux/MacOS students learned more about the `ssh` command and setting up the SSH `config` and `Host` profiles
- Key-based authentication for SSH. Students learned how to generate their own public/private key pairs, and how key-based authentication can be used to either enhance the security of their SSH connectivity through two-factor authentication (*something they have* – their SSH private key files, and *something they know* – the password required to decrypt those private key files) or to further enhance the convenience of remote access via SSH by no longer requiring any passwords at all. Students learned the benefits and risks that come with their decisions regarding private key security. Additionally, students also learned how to configure key-based authentication on the SIEM, IPS and Kali virtual machines – including how transfer the SSH public key to remote systems, as well as the particular location, and file permissions required for key-based authentication to function properly.
- How to enable SSH access as the `root` user. Students learned about the advantages and risks that come with allowing the `root` user to log in remotely using the SSH protocol as well as ways to limit that risk.
- How to disable password authentication over SSH entirely for the SIEM, IPS and Kali virtual machines, to better enhance the security of their lab environment.

Here is a list of tasks left for students to finish their lab environment:

- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in chapter 17, *Network Intrusion Detection*, starting on p. 980.
- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in chapter 18, *Setting up Splunk*, starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out chapter 19, *End of the Beginning*, starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out chapter 20, *Extra Credit*, starting on p. 1055.

Chapter 16 Patch Notes

-This chapter is probably the most heavily re-written out of all the chapters I've done so far.

-This chapter is considered to be a parallel to Chapter 18 from the first book. Some of the stuff from the beginning of the chapter, I've elected to split out into a separate chapter altogether, or incorporate into other chapters.

-A lot more detail has gone into the sections detailing how to set up either a bastion host virtual machine, or a physical bastion host system, using a raspberry pi running Raspbian as an example.

-A lot has changed with raspberry pi hardware. I cover the hardware required to use an old raspberry pi as a jump box, as well as how to use the newer raspberry pi hardware.

-The Raspberry pi imager is a thing now, and I no longer have to tell students to manually download IMG files, or SD card formatting software, the imaging software does all of that automatically now.

-It's also no longer necessary to expand the filesystem using `raspi-config`.

-Previously I just told students to persist static routes on the bastion host using `/etc/rc.local`. Now, I teach students how to persist static routes using `rc.local`, `netplan` (Ubuntu), or `dhcpcd` (Raspbian).

-Instead of having to configure firewall rules on the pfSense WAN interface for every IP address students want to serve as a bastion host to their lab environment, I've elected to have them create an IP address alias, `Bastion_Hosts`, and reconfigure the firewall rules on the WAN interface to where the Source is set to the `Bastion_Hosts` alias. This makes adding additional bastion hosts to the lab environment much easier, and much more scalable.

-Took some time to explain the TCP forwarding (SSH Tunnels) a bit more in-depth. I've documented the difference Forward, Reverse, and Dynamic SSH tunnels, along with diagrams that illustrate the function they provide.

-Linux users no longer use aliases containing SSH connection strings to connect to the bastion host or lab virtual machines, we use the `~/.ssh/config` file now. I show students how to create aliases for SSH connection strings, but only just.

-Windows users are given instructions on how to enable key-based authentication for WinSCP and are advised that if they disable password authentication over SSH later in the chapter, that WinSCP will only work by enabling key-based authentication.

-I've provided students with an entire section in this chapter to help troubleshoot key-based authentication, as well as SSH tunnel problems in general.

-The instructions on how to use Dynamic SSH tunnels and FoxyProxy to Proxy web traffic for the lab environment through the bastion host is, in my opinion, much better written.

-I've also taken the effort to illustrate how to install and configure FoxyProxy Standard on both Mozilla Firefox and Google Chrome, as well as illustrate the differences between the two.

-I've also provided a troubleshooting checklist for students who are experiencing problems with their dynamic tunnels and/or FoxyProxy.

- The instructions for enabling SSH authentication as the root user are a little bit clearer about some of the benefits vs. the risk enabling this access presents.

- As it turns out, disabling password authentication by setting `UsePAM` to `no` in `/etc/ssh/sshd_config` is a bad idea. Jeremi M. Gosney (@jmgosney), a former embedded Linux developer, and password cracking professional provided some guidance on best practices for disabling password auth over SSH, incorporated into this chapter.

Chapter 16: Routing and Remote Access for Bare-metal Hypervisors

The purpose of this chapter, as the name implies, is to help students establish consistent remote access to their lab environment hosted on the only bare-metal hypervisor covered in this book (so far), VMware ESXi. I felt that this subject needed to be covered *somewhat* separately because setting up consistent remote access to a bare-metal hypervisor lab is a little bit more involved than it is on hosted hypervisors. However, many of the concepts covered in Chapter 15, *Routing and Remote Access for Hosted Hypervisors*, will apply to this chapter as well. Students will need to establish static network routes, and configure remote access using the SSH protocol either via mRemoteNG or the ssh client on Windows or Linux/macOS workstations, respectively. Additionally, they will have the option of configuring key-based authentication, ssh access as the root user, and/or disabling password authentication over SSH, the same as chapter 15.

The major difference in this chapter, is that students will learn how about bastion hosts (sometimes referred to as a "jump box") and SSH tunneling. While this additional content does add a little bit of complexity, it also means consistent access to students' bare-metal lab environments in a manner that can easily accommodate additional students accessing the same resources.

Note: Please be aware, that much of the content covered in later sections of this chapter is practically identical, if not very similar to content covered in chapter 15, *Routing and Remote Access for Hosted Hypervisors*. When this occurs, students will be guided to refer to specific sections and pages of chapter 15 for detailed guidance. They will also be provided with a summarized list of tasks to perform and illustrations to provide visual cues on how to perform the task. Key tasks that are different due to the nature of bare-metal hypervisors will be covered as necessary.

16.1 A Brief Review: Bare-metal Hypervisors vs. Hosted Hypervisors

Way back in Chapters 3 and 4, students learned about the differences between Hosted and Bare-metal hypervisors, and how virtual network functions on both types of hypervisors:

Hosted hypervisors are generally installed as software on top of a desktop operating system. They typically feature network segments that grant virtual machines connected to them with unique network connectivity – NAT (share the hosts network connectivity, using Network Address Translation), Bridged (VMs are directly connected to the local network through the hypervisor host, but have their own IP and MAC address presence on the network), Internal (VM-to-VM network communication only), and Host-Only (VMs-to-VM communication and/or VM to Host).

Bare-metal hypervisors are installed as operating systems on dedicated hardware, with some form of network connectivity. A separate workstation is required to interact with the hypervisor and configure it. Networking is typically done with virtual switches that are either uplinked to the dedicated server's network interface (effectively "bridging" virtual machines connected to that virtual switch to the physical network the server is connected to), or they are not uplinked at all (in which, virtual machines connected to a virtual switch that is not uplinked are nearly identical to a hosted hypervisor's "Internal" network segments).

16.1.1 Lab Network Design on Hosted Hypervisors

Recall the design and layout of the lab environment (Jump back to Chapter 6, fig. 6-1, p.58). In Chapter 15, Students who chose to build their lab environment on a hosted hypervisor are taking advantage of the "Host-Only" virtual adapter (attached to the LAN network segment) in order to be able to connect to the SIEM, IPS and Kali virtual machines over SSH. In Chapter 14, students made necessary changes to the pfSense firewall to allow connectivity to the Kali Linux VM over SSH (port 22) in the IPS 1 segment, while in Chapter 15, they enabled the SSH service on the Kali VM, and configured a permanent (if not semi-permanent) static route to the IPS network, so that the physical hypervisor host knows where to send its packets to reach the IPS network (e.g., out of the host-only virtual interface, to the pfSense virtual machine's LAN network interface).

16.1.2 Lab Network Design on Bare-Metal Hypervisors

What about students who chose to configure their lab environment on a bare-metal hypervisor? There are no shortcuts. There are no clever tricks and taking advantage of "host-only" virtual adapters. That management workstation (e.g., your Windows, Linux, Unix-like or MacOS workstation) is considered an "external" network device. That "external" network device needs firewall rules in place on the WAN interface of the pfSense VM that will allow it access the virtual machines on both the LAN and IPS network virtual switches. This is why in both chapter 13 and 14, it was repeatedly pressed into students that static DHCP allocations or static IP addresses were extremely important for the following systems:

- The management/configuration interface IP address of the bare-metal hypervisor server itself
- The WAN interface of the pfSense VM
- The workstation students plan on using to access their lab environment

Students need the IP address of the server running ESXi to stay consistent in order to be able to log in and manage the virtual machines in the lab environment. The IP address of

the pfSense VM's WAN interface needs to remain consistent to use for static routing (more on this later), and to ensure consistent access to the WebConfigurator for modifying network access to the lab environment. Finally, the workstation students use needs a consistent IP address to ensure consistent access to the lab environment virtual machines. With all of this information in place, let me pose a couple of questions:

- What happens if the workstation students operate is on a network segment in which they cannot statically configure the IP address, or ensure a static DHCP allocation?
- What happens if and when students want to share access to their lab with their peers or co-workers?

There's really no easy answer to the first question, and while the second problem could be resolved by adding more firewall rules that allow the IP addresses of co-workers' or peer IP addresses (or setting up pfSense aliases for the firewall rules on the WAN interface), it's not an easily sustainable solution. With these scalability and sharing problems in mind, let's talk about bastion hosts.

16.2 Introduction to Bastion Hosts

While there are numerous ways to try and address the scalability and consistent access issues, setting up a bastion host is probably one of the easier and better ways to do so. A bastion host, sometimes referred to as a "jump box" or "redirector", is an intermediate system that users connect to in order to access other resources that the bastion host has access to. Bastion hosts both provide enhanced accessibility to students' bare-metal lab environments, as well as enhanced security. Instead of requiring multiple static DHCP entries and multiple (or more complex) firewall rules to allow them all access, configure an additional host on the physical network (either an old desktop, small embedded system, or even an additional virtual machine on the bridged segment/virtual switch), grant it a static IP address or DHCP allocation, and use the bastion host to establish connections to the rest of the lab virtual machines. It's a single point of entry, and can have additional auditing, logging and security measures placed on it to provide enhanced security. Bastion hosts are used extensively in several regulatory environments for these purposes.

How another host going to solve all these problems? The SSH protocol has a well-known feature called TCP forwarding (also referred to as SSH tunneling). In a nutshell, TCP forwarding allows students to treat the bastion host like a proxy. This enables any student, from any IP to connect to the bastion host (so long as they have credentials that work), configure TCP forwarding, and access the lab VMs as needed (again, with valid credentials). Because TCP forwarding makes the connection appear to be coming from the bastion host, there's no need to set up additional firewall rules, so long as the bastion host is allowed to access the lab environment over the network.

What if I Don't Need to Share?

Some of you may be fortunate to have access to server hardware, and have it set up in some forgotten (or hopefully soundproofed) portion of your home. You don't really have a need to share your bare-metal lab with anyone, and you can set up static DHCP allocations for your workstation as necessary. Why should I bother setting up a bastion host?

Technically, you don't have to. Chapter 13 and Chapter 14 repeatedly drilled into you the importance of having a static IP addresses configured for the ESXi web interface, WAN interface of the pfSense VM, and the workstation you're doing all your changes from. In fact, the firewall policy in chapter 14 is already set up to allow access to the entire lab environment from your management workstation. All you'd need is to configure static routes to tell your management workstation how to get to the LAN and IPS networks, using the `route` command for your workstation's operating system. This chapter covers the correct routes to add for Linux (since that's the operating system the jump box will be using), but for MacOS and Windows, they won't be covered here. You'll need to get creative, and adapt the Linux route commands in this chapter with the syntax demonstrated for MacOS or Windows route in Chapter 15.

While you don't *have to* set up a bastion host, *you really should*. Learning how to set up and use a bastion host teaches you how to creatively pivot from a resource you have access to, to resources you want further access to. This is something that most IT professionals can appreciate but for different reasons.

16.3 Creating A Bastion Host

In this section, students will go through the process of creating their own bastion host for use with their bare-metal lab environment. There will be two options presented:

- Creating an additional virtual machine attached to the Bridged virtual switch/port group on the VMware ESXi server
- Creating a bastion host using spare physical hardware – This option features configuring a Raspberry Pi computer as a physical jump box students may use as a bastion host for their lab environment, instead

Note: If students don't want to host their jump box on the ESXi server due to a lack of resources (e.g., RAM, disk space, etc.), and do not have access to a Raspberry Pi, any old or spare hardware you have available capable of running Ubuntu Server and OpenSSH SSHD (that can be configured with a static IP address or static DHCP mapping) will fit the bill.

The installation instructions for Ubuntu Server or more less exactly the same for physical hardware as they are for a virtual machine. Follow the guidance provided for the SIEM virtual machine in chapter 13, *section 13.7.2.1, pp-603-610*. The illustrations 13-63 through 13-67 can also be utilized for general installation guidance as well.

As a general reminder, here is a list of tasks students should have performed for their bare-metal hypervisors labs so far:

- Installed VMware ESXi and applied the free license key
- Configured Datastores to hold operating system ISOs, and virtual machines
- Performed necessary network configurations for the lab environment:
 - Configured the Bridged, LAN, IPS 1 and IPS 2 Virtual Switches and their corresponding Port Groups
 - Created and performed initial setup of the pfSense, SIEM, IPS, Kali, and metasploitable 2 virtual machines
 - Configured pfSense network services (DHCP, SQUID, NTP, DNS) and Firewall Policies for the WAN, LAN and OPT1 networks
 - Configured either static IP addresses or static DHCP assignments for the ESXi web interface, WAN interface of the pfSense VM, and are capable of adding one more static DHCP allocation for the bastion host we will be creating together

The remainder of this chapter assumes that all of this groundwork has been laid. If you have not finished doing so yet, refer back to chapters 13 and 14 to finish setting up the lab environment, to achieve all of the tasks listed.

16.3.1 Creating a Bastion Host Virtual Machine on VMware ESXi

By now, students should be pretty familiar with the process for creating virtual machines on VMware ESXi. Below is a spec sheet, and a screen capture of the summary page of the *New Virtual Machine Wizard* to help guide students through the process of creating the bastion host virtual machine:

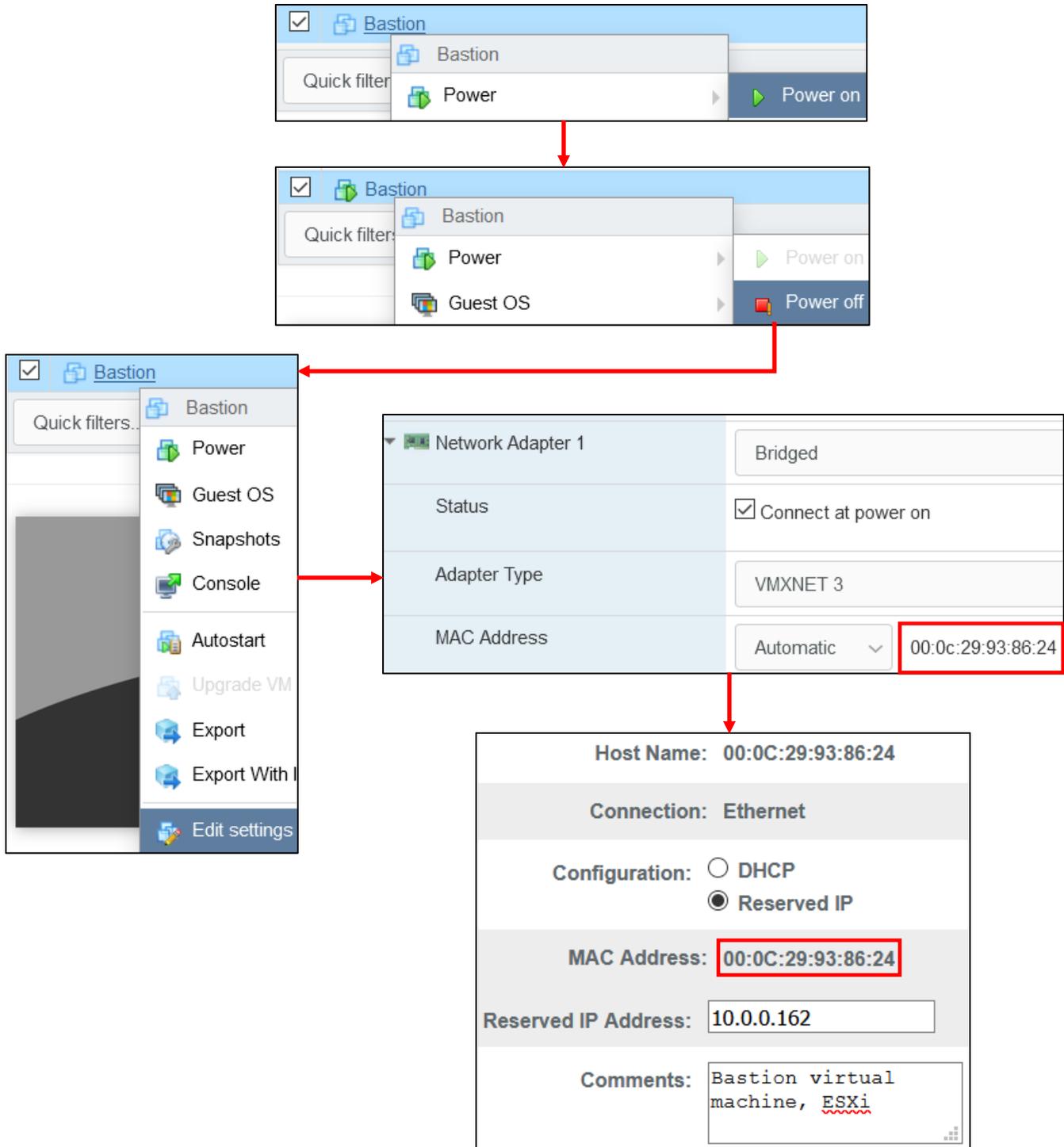
Name:	Bastion
Guest OS	Linux
Family:	
Guest OS	Ubuntu Linux (64-bit)
Version:	
Storage:	Use any available datastore
Memory:	1GB (1024MB), or more
Hard Disk 1:	30GB Thick Provisioned, eagerly zeroed (or more)
Number of Network Adapters:	1
Port Groups:	Network Adapter 1: Bridged
CD/DVD Drive 1:	Select <i>datastore ISO file</i> . Locate the Ubuntu Server 20.04 ISO, and select it.
Other:	Remove <i>USB controller 1</i>

Name	Bastion
Datastore	Mune
Guest OS name	Ubuntu Linux (64-bit)
Compatibility	ESXi 7.0 virtual machine
vCPUs	1
Memory	1024 MB
Network adapters	1
Network adapter 1 network	Bridged
Network adapter 1 type	VMXNET 3
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SCSI controller 0	LSI Logic Parallel
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	30GB
Datastore	[Mune] bastion/
Mode	Dependent
Provisioning	Thick provisioned, eagerly zeroed
Controller	SCSI controller 0 : 0
CD/DVD drive 1	
Backing	[Flash_Step] ISOs/ubuntu-20.04.1-live-server-amd64.iso
Connected	Yes

16-1: By now, students should be familiar with the process for creating new virtual machines on ESXi. Here is the summary page for the bastion host virtual machine. To summarize, allocate 1 virtual CPU, 1GB+ of RAM, and 30GB+ of disk space – thick provisioned, eagerly zeroed. Attach the Ubuntu Server 20.04 ISO, and remove the USB controller. Ensure only one network adapter is allocated to this virtual machine, and that it is attached to the *Bridged* port group.

With the virtual machine created, power it on, then power it back off immediately (Disregard the pop-up that warns about potential data loss—there's literally no data on the virtual machine to lose yet.). As students discovered in chapter 13, these actions force ESXi to assign a MAC address to the network interfaces on the VM. With the VM powered off again, select *Edit settings*, and record the MAC address of *Network Adapter 1*.

Students will to create a static DHCP mapping for the bastion host virtual machine, using the MAC address they just recorded. It is absolutely vital that the bastion host's IP address never change. Unfortunately, I have no means on instructing students how to do this, as every network is different and ever piece of network equipment has different instructions for configuring static DHCP mappings. I recommend referring to the documentation for the network equipment for guidance on how to do so. If the lab network is being configured on an enterprise or office network, students may be required to work with network or system administrators in order to reserve a static DHCP allocation.



16-2: Power the VM on, then off again immediately in order to force ESXi to assign a MAC address to *Network Adapter 1*. Record that MAC address, and use it to create a static DHCP mapping. Unfortunately, I have no idea what equipment students use on their home networks, and every vendor has a completely different method for implementing static DHCP mappings (if they support it at all). Students should refer to documentation for their network equipment for guidance on completing this task. For enterprise/office networks, coordination with IT or Network administrators may be necessary to perform this task.

With initial VM setup, and static DHCP allocations out of the way, the next step is operating system installation. Ubuntu Server is going to be the operating system of the bastion host VM, and by this point, students should be mostly familiar with the various stages of the installer. In general, the installation settings are nearly identical to those of the SIEM virtual machine, so if a refresher is required, refer to Chapter 13, sections [13.7.2.1](#), [13.7.2.2](#), and [13.7.2.3](#), pp. 603-615 for further guidance. Here are the important tasks, and differences to consider:

- On the *Network Connections* screen, verify that the static DHCP allocation created for the bastion host on the local, physical network was successfully applied. If students need to set a static IP address manually, check out the sidebar conversation, *Doing Things Manually*, down below.
- Unless the physical network the bastion virtual machine is connected to requires the use of a proxy for internet access, the *Proxy Address* input box on the *Configure proxy* screen can remain blank
- On the *Profile setup* screen, be sure to save the username and password for the bastion host to a password manager, and to set the *Server name*.
- **On the SSH Setup screen, make absolutely sure to check the Install OpenSSH server checkbox.**
- Once the operating system is installed, be sure to remove the *CD/DVD Drive*, and *SATA Controller 0*.
- Log in to the bastion host's virtual console. Verify network connectivity using the same troubleshooting commands we used for the SIEM, IPS and Kali VMs:

```
ping -c 4 www.google.com
nslookup www.google.com
curl -I https://www.google.com
```

- Assuming network connectivity is working properly, run:

```
sudo su -
apt-get update
apt-get -y dist-upgrade
init 6
```

- Create a snapshot of the VM after it is finished rebooting.

Note: If you're keeping track of your lab virtual machines, update your asset inventory to include details about your new bastion host virtual machine

When finished, proceed to [section 16.4](#) (p. 882) to learn how to utilize your new bastion host.

```
Network connections
Configure at least one interface this server can use to talk to the network and which preferably provides sufficient access for updates

NAME      TYPE  NOTES
[ ens160  eth  - ]
DHCPv4    10.0.0.162/24
00:0c:29:93:86:24 / VMware / VMXNET3 Ethernet Controller
```

```
Configure proxy [ Help ]
If this system requires a proxy to connect to the internet, enter its details here.
Proxy address: _____
```

```
Profile setup [ Help ]
Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: ayy _____
Your server's name: bastion _____
The name it uses when it talks to other computers.
Pick a username: ayy _____
Choose a password: *****
Confirm your password: *****
```

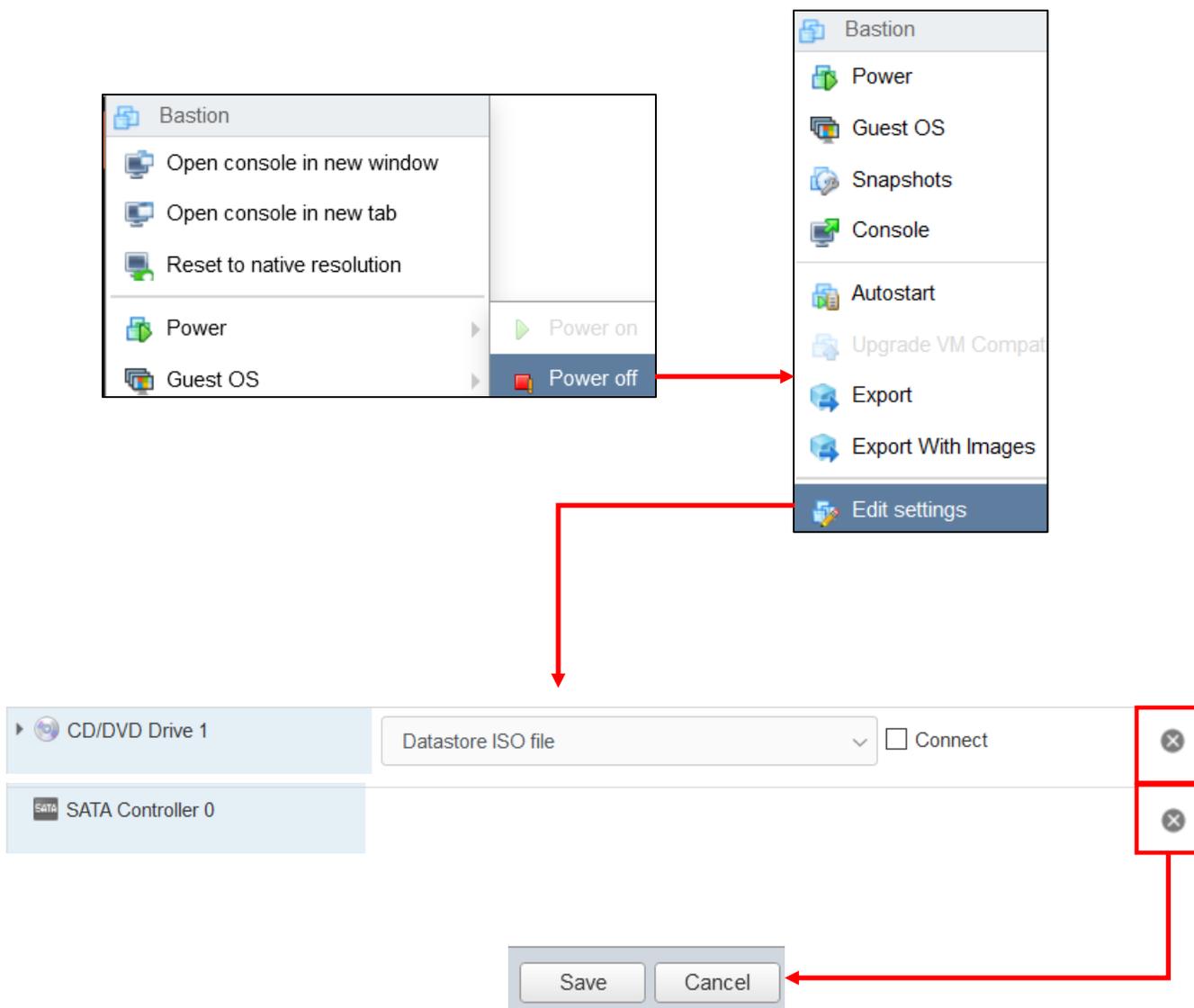
```
SSH Setup [ Help ]
You can choose to install the OpenSSH server package to enable secure remote access to your server.

[ X ] Install OpenSSH server
```

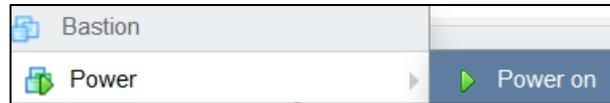
16-3: Welcome back to the Ubuntu Installer. Students will be installing Ubuntu Server 20.04 using the exact same method used for installing it on the SIEM VM in chapter 13, with exception to the following settings:

- On the *Network Connection* screen, students should confirm that their static DHCP allocation has been applied successfully
- On the *Configure proxy* screen, the Proxy address input box can remain blank, unless students utilize a proxy configuration on their local network
- On the *Profile setup* screen, be sure to record the username and password for the new virtual machine to a password database or similar safe location.
- On the *SSH Setup* screen, **it is extremely important that the Install OpenSSH server checkbox is checked**

Proceed through the rest of the installation process as normal.



16-4: Once the Ubuntu installer has completed, *Power off* the bastion host VM, and enter the *Edit settings* menu. Remove *CD/DVD Drive 1*, Followed by *SATA Controller 0*. Recall that ESXi can be very finicky if users attempt to delete both pieces of virtual hardware at once. If this happens, remove *CD/DVD Drive 1* virtual hardware, click *Save*, enter the *Edit settings* menu again, remove *SATA Controller 0*, and click the *Save* button again.

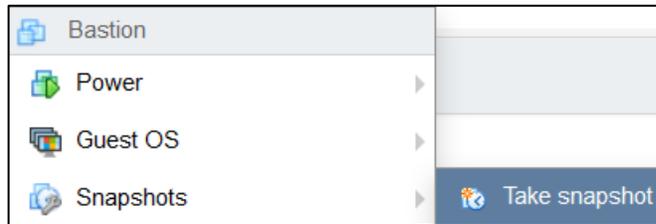


```
ayy@bastion:~$ ping -c 4 www.google.com
PING www.google.com(ord38s09-in-x04.1e100.net (2607:f8b0:4009:816::2004))
64 bytes from ord38s09-in-x04.1e100.net (2607:f8b0:4009:816::2004): icmp_seq=0 ttl=64 time=22.637 ms
64 bytes from ord38s09-in-x04.1e100.net (2607:f8b0:4009:816::2004): icmp_seq=1 ttl=64 time=24.856 ms
64 bytes from ord38s09-in-x04.1e100.net (2607:f8b0:4009:816::2004): icmp_seq=2 ttl=64 time=28.279 ms
64 bytes from ord38s09-in-x04.1e100.net (2607:f8b0:4009:816::2004): icmp_seq=3 ttl=64 time=2.107 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 22.637/24.856/28.279/2.107 ms
ayy@bastion:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.9.36
Name:   www.google.com
Address: 2607:f8b0:4009:816::2004
ayy@bastion:~$ curl -I https://www.google.com
HTTP/2 200
ayy@bastion:~$ sudo su -
[sudo] password for ayy:
root@bastion:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
root@bastion:~# apt-get -y dist-upgrade
Processing triggers for initramfs-tools (0.136ubuntu6.3) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-66-generic
root@bastion:~# init 6_
```

16-5: Power the bastion host VM back on, and log in via the virtual console. Use the ping, nslookup, and curl commands to verify internet connectivity. Afterwards, become the root user (sudo su -), then use apt-get update, and apt-get -y dist-upgrade to apply the latest software updates. Once finished, use either init 6 or reboot to restart the virtual machine.



Take snapshot for Bastion

Name	bastion host baseline snapshot
Description	Ubuntu 20.04 installed <u>OpenSSH</u> installed network connectivity checks passed static <u>DHCP</u> mapping confirmed updates applied

Snapshot the virtual machine's memory.

Quiesce guest file system (needs VMware tools installed).

Take snapshot

16-6: last but not least, take a snapshot of the bastion host VM to use as a baseline known good configuration students can restore to, if necessary. Congratulations, students now have a functional bastion host VM. Jump to section 16.4 to learn how to use it.

Doing Things Manually

Note: If you don't have a DHCP server on your local network, access *Edit IPv4* settings, on the *Network Connections* screen, then select *Manual* from the *IPv4 Method* drop-down menu. Manually enter the *Subnet*, *Address*, *Gateway* and *Name servers*, then *Save* your settings **You will need these settings to establish internet access.**

NAME	TYPE	NOTES	
[ens160	eth	-	▶ (close)
DHCPv4 10.0.0.162/24			Info
00:0c:29:93:86:24 / VMware / VMX			Edit IPv4

Edit ens160 IPv4 configuration

IPv4 Method:

Subnet:

Address:

Gateway:

Name servers:
 IP addresses, comma separated

Search domains:
 Domains, comma separated

[Save]
 [Cancel]

16-7: If you need to manually set a static IP address for your bastion host VM, Access the *Edit IPv4* settings menu on the *Network Connections* screen, then select *Manual* on the *IPv4 Method* drop-down. From there, Fill out the *Subnet*, *Address*, *Gateway* and *Name servers* input boxes. Unfortunately, I can't tell you the correct information to enter, but chances are, if you've had to set static IP addresses before, you probably know (at least better than I would) what needs to go in these fields.

16.3.2 Creating a Raspberry Pi Bastion Host

Instead of a virtual machine, students may use any physical system as a bastion host to share access to their lab environment. For this section, we will be utilizing a Raspberry pi computer (from here on out, I'll be using RPI for short). Why? Well, they're ubiquitous, affordable and require little in the way of power in order to operate them. As stated earlier however, if students have other spare hardware available, they are more than welcome to use that for a physical bastion host, instead.

16.3.2.1 Prerequisites

There are plenty of versions and revisions of the RPI, but the B or B+ versions are my personal recommendation, because they feature a wired network interface, for more reliable network connections. Depending on what hardware students are using for their lab environment, at a minimum they will need the following items in order to set up and use their raspberry pi as a bastion host:

- **SD (or microSD) card with at least 32GB of space**
 - The official documentation states users can get away with using 16GB SD cards, to SD cards with as little as 4GB of space, but 32GB is considered pretty standard fare, and can be found for ~10.00 USD at most electronics retailers.
 - Newer RPI models (2014 or later) use microSD cards. The 32GB size recommendation still applies (and somehow, microSD cards are even cheaper than standard SD cards).
 - Depending on the desktop hardware students are using, an SD card reader (and/or a microSD to SD card adapter) may be necessary.
- **USB 2.0 keyboard**
 - Students will need to input commands using a keyboard + monitor in order to enable remote access. Once remote access is enabled, keyboard/monitor will no longer be required.
- **Monitor capable of supporting HMDI output + an HDMI cable (and necessary adapters)**
 - Some of the newer RPI models use micro HDMI ports and will require a micro HDMI to HDMI adapter to display video.
 - Once remote access is enabled, keyboard/monitor will no longer be required.

- **USB power cable + power source**

- Older RPI models use a micro-USB to USB cable to provide power.
 - Official instructions recommend using a wall outlet adapter to provide power. If students have a wall adapter from an old cell phone, these will usually work in a pinch.

Note: The older RPI B/B+ boards can *technically* be powered off of the USB rail of another system. For example, my ISP-provided router has USB ports I have used to power my old model B to operate as a jump box. Please note that while this method is *possible*, it can be *unreliable*. If done incorrectly, it could potentially result in damaging both your equipment as well as the RPI board.

- Newer RPI boards use USB-C to USB cables to provide their power. This is because the newer RPI boards have increased power requirements, increased resources and higher performance, multi-core CPUs.
 - The Raspberry Pi Foundation sells an official USB-C wall adapter that can be used to power newer boards. However, if students have a USB-C + Wall adapter from an old cell phone, these will usually work just as well.
- **Network cable (RJ-45) + connection to physical network**
 - Students should ensure they have a spare network cable, and active network port available in order to connect their bastion host to their network.



16-8: In the first picture on the top of this page, is pictured an RPI 4 Model B (1) along with a microSD card and adapter to fit the microSD card into a standard SD card slot (2), a micro HDMI to HDMI adapter to output video (3), and finally the USB-C wall adapter for power (4). In the second picture below, is an early generation RPI Model B+ (5), along with an SD card (6), and micro-USB wall adapter (7). Students will need these items, plus an HDMI cable, network cable, USB keyboard, and monitor (not pictured) to do the initial installation and setup on the RPI.

Custom Upgrade Kit

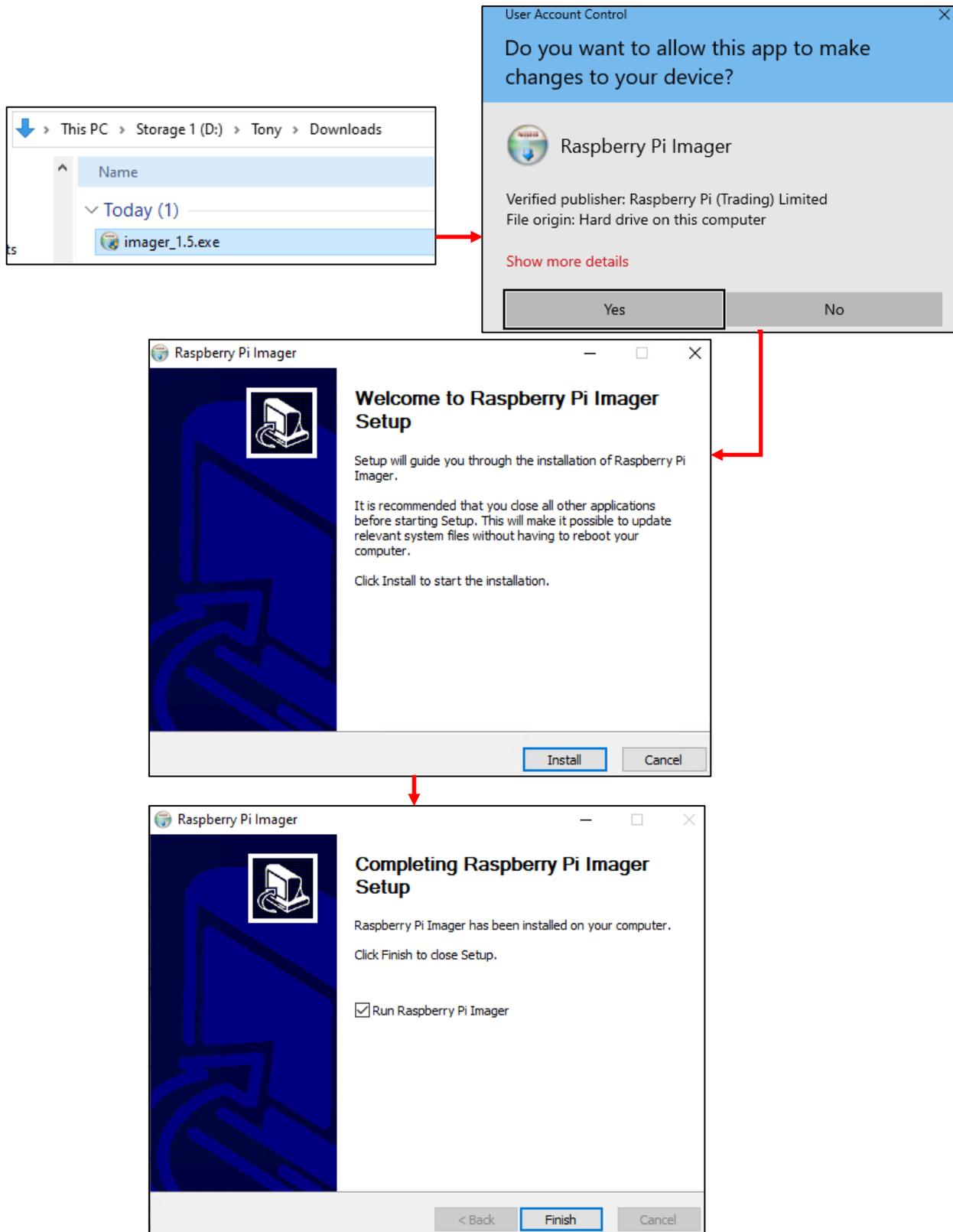
Are you concerned about the number of converters and adapters needed to get things up and running? Fortunately, there are retailers that provided Raspberry Pi "starter kits" that include everything you need to get started, including the board itself. In my online searches, the most popular starter kits I've found are provided by CanaKit. If you're interested, check out <https://www.canakit.com>. I would recommend getting one of the raspberry pi 4 starter kits. They include a microSD card, an SD card adapter for your PC, a USB-C power adapter, a micro HDMI to HDMI cable, and a protective case for the RPI with heatsinks and a fan. The prices I've seen for this combo range from 80.00 USD (2GB model) to 120.00 USD (8GB model).

16.3.2.2 Raspberry Pi Imager

The next step involves formatting the SD (or microSD) card, and installing Raspbian. Raspbian is a special branch of Debian Linux specially developed and maintained for Raspberry Pi hardware. The process for formatting the SD card and installing Raspbian has been greatly simplified through a software package called *Raspberry Pi Imager*.

16.3.2.2.1 RPI Imager Installation Instructions: Windows

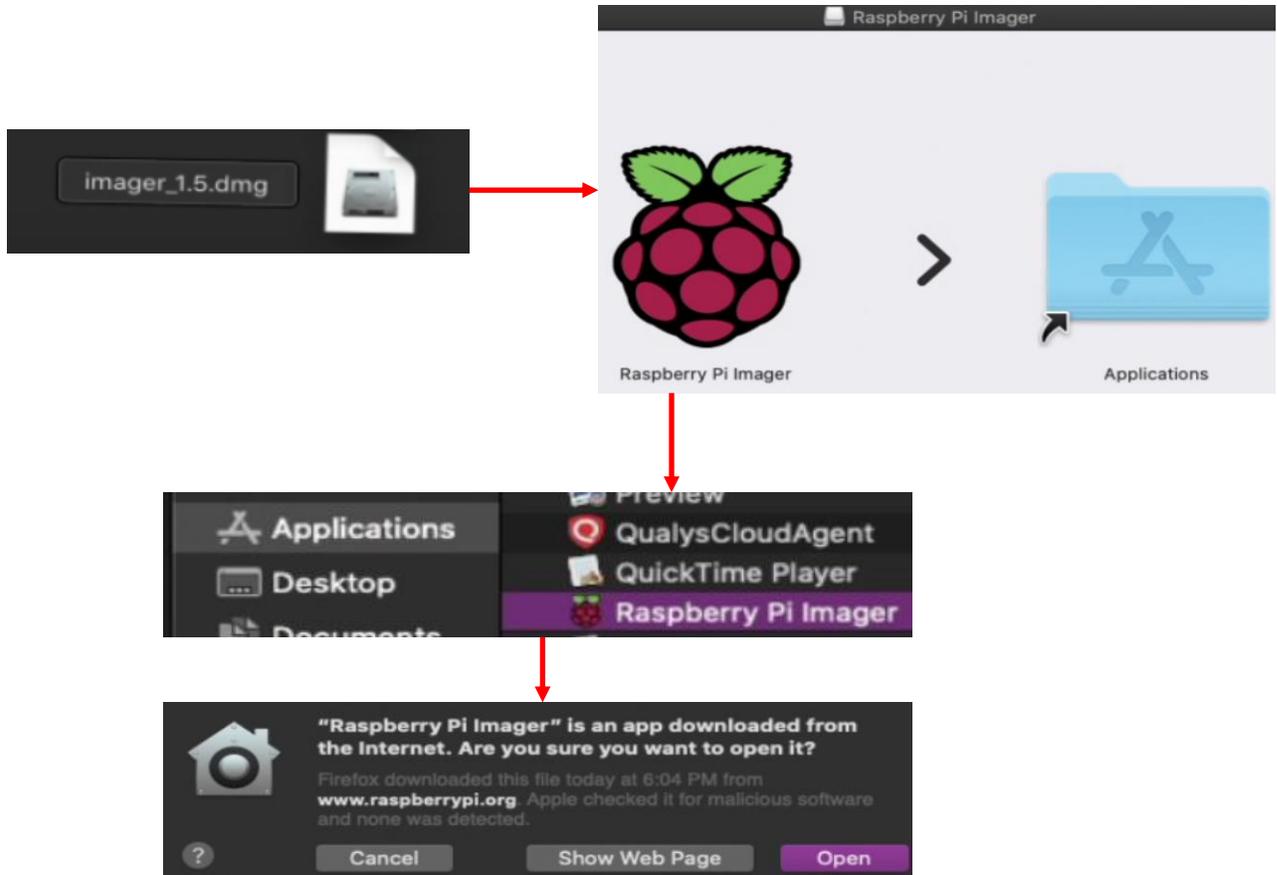
Download RPI Imager from <https://www.raspberrypi.org/software>. When finished, double-click the `imager_x.x.x.exe` executable. If UAC is enabled, students will be prompted to allow the executable to make changes to the system. Select *Yes* to continue. A window labeled *Raspberry Pi Imager* appears. Click the *Install* button, and the installation will complete automatically. When the installer is finished, ensure the *Run Raspberry Pi Imager* checkbox is checked, then click the *Finish* button to exit the installer.



16-9: Installing the Raspberry Pi Imager software on Windows is as simple as starting the installer executable, accepting the UAC prompt, Clicking the *Install* button, then clicking *Finish*. By default, the *Run Raspberry Pi Imager* checkbox should be checked, but if it is not, make sure it is checked, or students will need to need to locate and start the Raspberry Pi Imager application manually.

16.3.2.2.2 RPI Imager Installation Instructions: MacOS

Download RPI Imager from <https://www.raspberrypi.org/software>. When finished, double-click the `imager_x.x.x.dmg` in Finder. The Raspberry Pi Imager dmg archive opens. Drag the giant Raspberry Pi Imager icon to the Applications directory icon. When finished, open the Applications directory in Finder, and double click on the Raspberry Pi Imager application. A pop-up will appear asking if students are sure they wish to continue. Click the *Open* button to proceed.



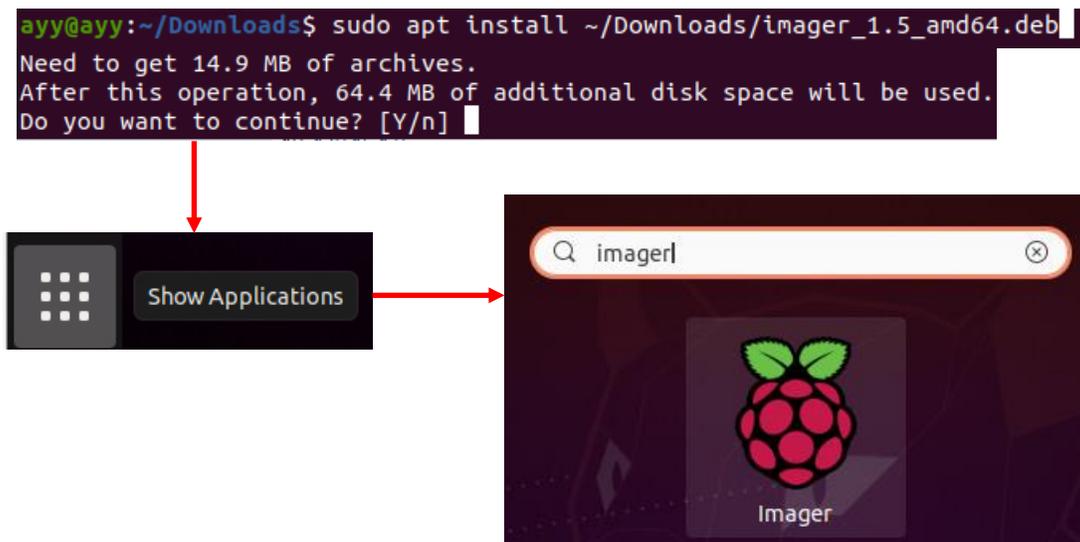
16-10: Download and double-click on the Raspberry Pi Imager dmg file, then drag the giant raspberry over to the *Applications* folder. Open the *Applications* folder in *Finder*, and double-click on Raspberry Pi Imager application. Students will be greeted with a prompt asking them to confirm they wish to run the application. Click *Open* to proceed.

16.3.2.2.3 RPI Imager Installation Instructions: Ubuntu Desktop 20.04

Download RPI Imager from <https://www.raspberrypi.org/software>. When finished, open your preferred terminal application and navigate to `~/Downloads`, where the `imager_x.x.x_amd64.deb` file should be located and run the following command:

```
sudo apt install ./imager_x.x_amd64.deb
```

Students will be prompted on whether or not they want to install the missing dependencies or software the `rpi-imager` application requires to run. Enter `Y` for yes, and hit enter to continue. Once finished, click the *Show Applications* button in the lower corner of the Ubuntu window manager, and type the word `imager`, in the search bar. Click on the large raspberry icon to start the Raspberry Pi Imager application.



16-11: Download the `imager_x.x_amd64.deb` file and run the following command:

```
sudo apt install ./imager_x.x_amd64.deb
```

Students will use the `apt` command to install the `rpi-imager` software package, and resolve software dependencies, or additional software it needs to function. Afterwards, open the *Show Applications* menu, search for `imager`, and click on the giant raspberry icon to open Raspberry Pi Imager.

RPI-Imager on other Linux Distros

Most of you out there with technical Linux chops would probably hazard a guess that if the Linux installer is distributed as a .deb file, then there's a good chance the rpi-imager package could be installed on other .deb-based Linux distributions. For example, maybe Kali, or Debian itself? *Potentially*. I myself have not tested installing the tool Debian, but confirmed that the installation process described for Ubuntu will work on Kali Linux as well. If you're inclined to tinker with Linux software in order to bend it to your will, give it a try.

16.3.2.3 Installing Raspbian using Raspberry Pi Imager

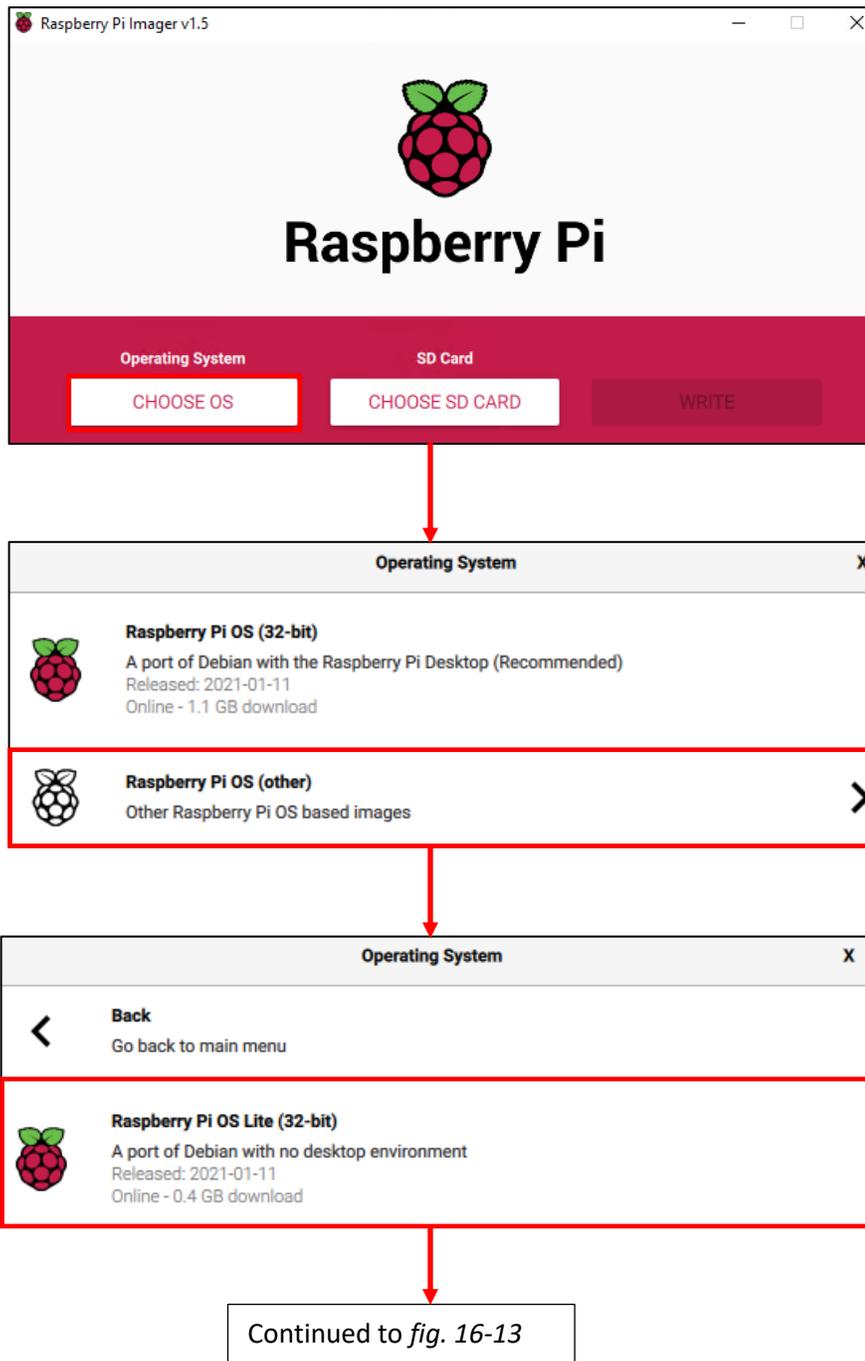
With the installation of Raspberry Pi Imager (henceforth referred to as RPI Imager) completed, students will now use the application to format their SD (or microSD) cards, and install Raspbian. Begin by inserting the SD card into your workstation, followed by opening the RPI Imager application.

Note: The interface for RPI Imager is consistent across different operating systems. This means that the interface and instructions for using RPI Imager should be identical regardless of what operating system students are using. I will be demonstrating how to use RPI Imager using Windows as my desktop operating system, but users on macOS and Linux should have no problem following along.

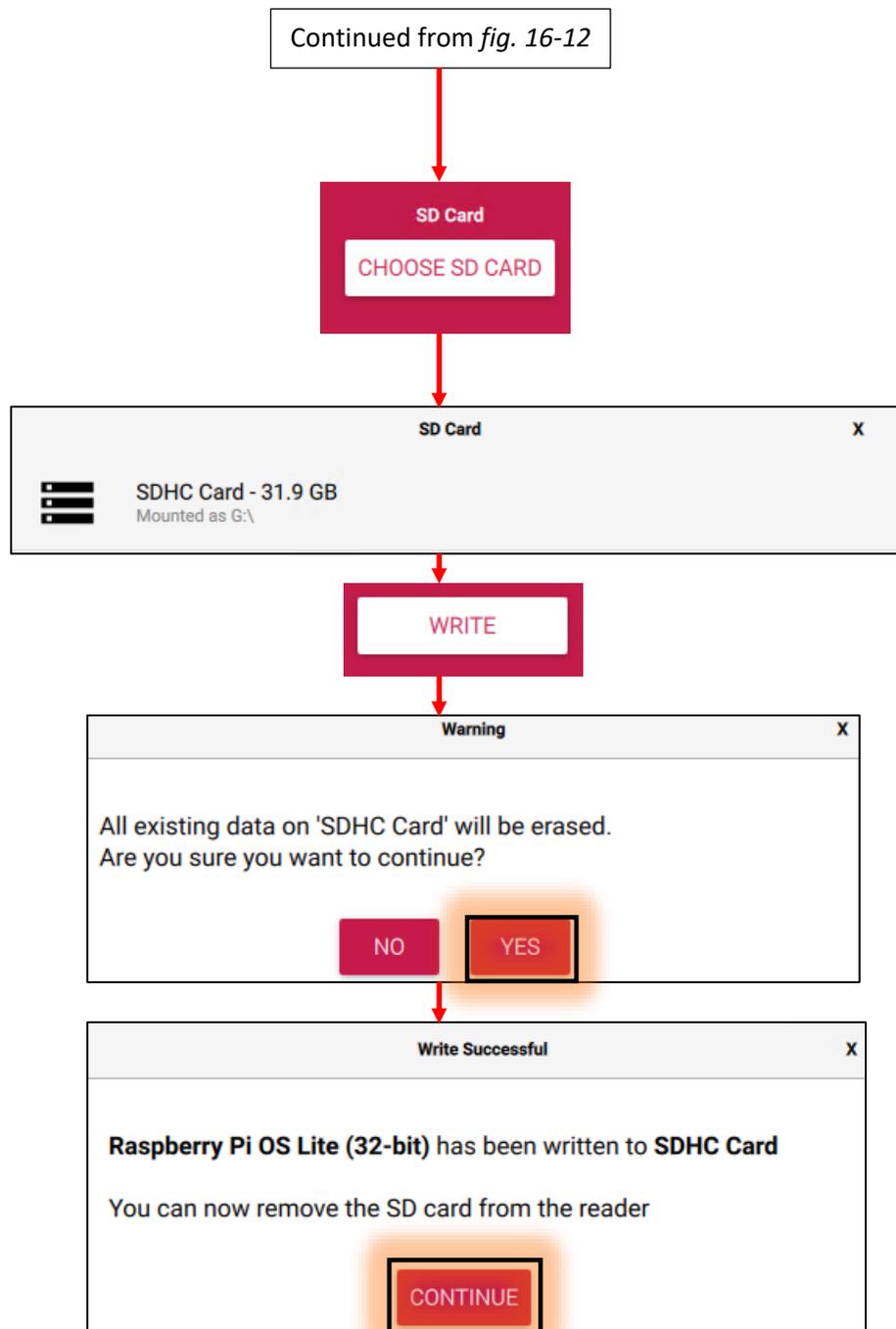
Students will be greeted with a window titled *Raspberry Pi Imager vX.X.X*. In the bottom portion of the window, are three buttons, two labeled *Operating System (CHOOSE OS)*, *SD Card (CHOOSE SD CARD)*, and a greyed out (for now) *WRITE* button. Click the *CHOOSE OS* button first. A new window titled *Operating System* appears. Click on the menu option labeled *Raspberry Pi OS (other)*, and on the screen that appears, select *Raspberry Pi OS Lite (32-bit)*. Selecting this option will return students back to the main window.

Next, click on the *CHOOSE SD CARD* button. Another window pops up with a list of SD cards the application was able to detect. Select an SD card to be returned back to the main screen.

Finally, Click the *WRITE* button to write the chosen operating system to the chosen SD card. Students will be given a final warning confirm that all data on the SD card will be erased in the process. Click *YES* to continue. When finished, the RPI Imager application will display a notification message that it's safe to remove the SD card. Remove the SD (or microSD) card, and install it on to the Raspberry Pi hardware.



16-12: Open the RPI Imager application, and begin by clicking the *CHOOSE OS* button. A menu labeled *Operating System* will appear. Click the option labeled *Raspberry Pi OS (other)*, then on the sub-menu that appears, select *Raspberry Pi OS Lite (32-bit)*.



16-13: Next up, Click the *CHOOSE SD CARD* button, then in the menu labeled *SD Card*, students will select the SD card they wish to use to install Raspbian. Finally, with the operating system and SD card selected, Click the button labeled *WRITE*. A warning appears notifying students that all existing data on the SD card will be lost. Select *Yes* to proceed. After some time passes, the *Write Successful* dialogue box appears to inform users it is safe to remove the SD card.

Offline Mode

Some of the more observant students may have noticed:

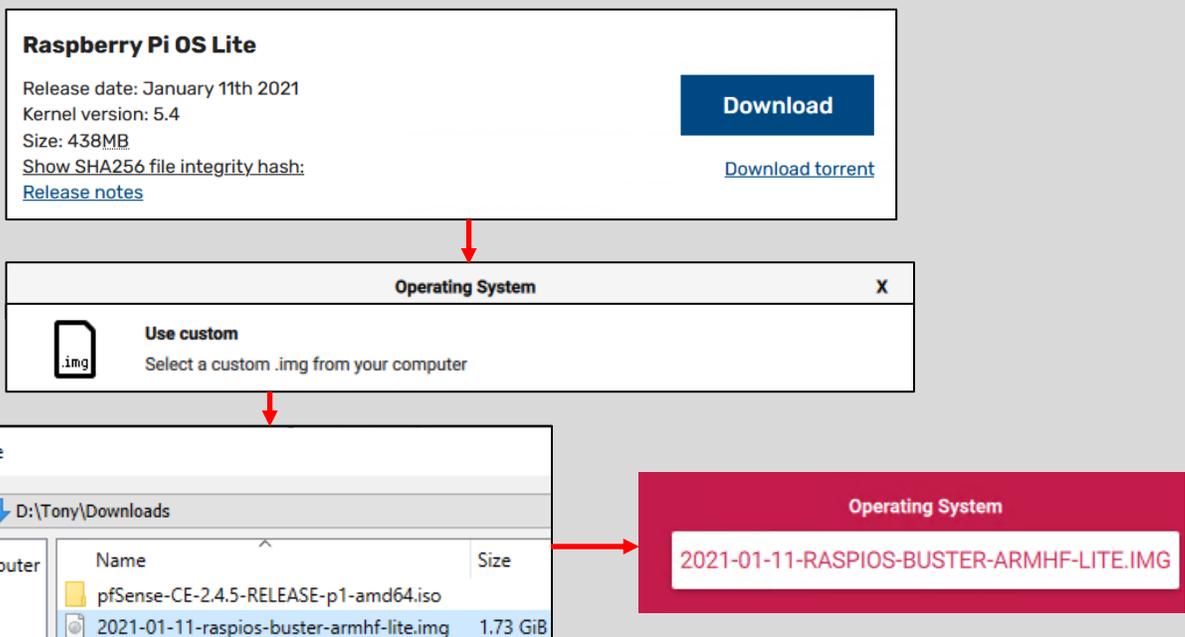
Online - 0.4GB download

In light grey text under the *Raspberry Pi OS Lite* operating system choice. This means exactly what you think it means: RPI Imager requires internet access to download the operating system we're installing to the SD card. Are you working on a network in which internet access is limited? Try this instead:

On a workstation with internet access, navigate to:

<https://www.raspberrypi.org/software/operating-systems/>

Locate the entry *Raspberry Pi OS Lite*, and download the ZIP file. Use your operating system's compression utilities (e.g., 7-zip, Finder, zip/unzip) to unzip the zip file. In its place will be an IMG file. Use whatever methods are required to "sneakernet" that decompressed IMG file over to your workstation without internet access. Start RPI Imager, and click the *CHOOSE OS* button, except this time, scroll all the way to the bottom of the *Operating System* menu and select the *Use custom* option. A file browser will appear. Browse to the location of the IMG file you just copied over, and use that as your OS instead. From here, follow the same process outlined above: Select the SD card to copy the image to, then click the *WRITE* button to begin the process.



16-14: If you don't have internet access, download (and decompress) the Raspberry pi OS Lite image from a computer that does, then select the *Use custom* option from the *Operating System* menu in the RPI Imager. Follow the rest of the instructions to install Raspbian to your SD card like normal.

16.3.2.4 Booting the Raspberry Pi and Configuring Raspbian

With the SD card flashed with the Raspbian Lite operating system, plug a keyboard into one of the available USB ports, the monitor into the (micro) HDMI port, network cable, (micro) SD card, and USB power cable into the raspberry pi (in that order) to boot the Raspberry Pi board. The boot process will take some time to complete, but be patient. After some time, students will be prompted to log in. **The default username for Raspbian is pi, and the default password is raspberry.** The first thing students should do upon logging in, is change the password for the pi user, because the default password for Raspbian is well-known, and leaving that default password enabled will make the bastion host vulnerable to password guessing. Run the `passwd` command. Students will be prompted to enter the password for the pi user, then prompted twice to enter the new password, followed by confirming that password. As always, document the new password for the pi user in a secure location, or a password manager.

Next, run the following commands:

```
sudo su -
raspi-config
```

Students need to become the root user in order to run `raspi-config`, a special command-line configuration menu for Raspbian that allows users to easily enable or disable a host of operating system features. Using the arrow keys on the keyboard, highlight *Interface Options*, and press enter. In the sub-menu that appears, highlight the *SSH* menu option, and hit enter on the keyboard again. A window will appear with the text:

```
Would you like the SSH server to be enabled?
```

Caution: Default and weak passwords are a security risk when SSH is enabled!

Use the arrow keys to highlight `<Yes>`, then hit enter once more to enable the SSH service. After a moment or two, another window will appear with the text:

```
The SSH server is enabled
```

Press the enter key to close this window, and return back to the `raspi-config` main menu. From here, students can press the tab key on their keyboard to highlight the `<Finish>` option, and hit enter to exit the menu. Next, type the `exit` command to exit the root shell.

The final thing students need to do is recover the MAC address of the raspberry pi's network interface to create a static DHCP mapping. The simplest way to do this is to run the command `ip addr`. Ignore the output for the `lo` interface, and instead focus on the remaining interface. In my case, the interface was named `eth0`. The MAC address is series of letters and numbers next to the field labeled `link/ether`. For example, my old model B+ has a MAC address of `B8:27:EB:B3:77:a6`.

Note: Some of the newer raspberry pi models might have the interface wlan0 for wireless connectivity. Ignore it and focus on the eth0 interface. Disregard wi-fi; acquire ethernet.

```
pi@raspberrypi:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether dc:a6:32:47:3f:a8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.138/24 brd 10.0.0.255 scope global dynamic noprefixroute eth0
        valid_lft 172551sec preferred_lft 150951sec
    inet6 2601:408:502:c330::204c/128 scope global dynamic noprefixroute
        valid_lft 604543sec preferred_lft 604543sec
    inet6 2601:408:502:c330:8265:a541:b304:d5fe/64 scope global dynamic mngtaddr noprefixroute
        valid_lft 188315sec preferred_lft 188315sec
    inet6 fe80::2b6:10ff:2f8c:3891/64 scope link
        valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether dc:a6:32:47:3f:a9 brd ff:ff:ff:ff:ff:ff
pi@raspberrypi:~$
```

16-15: Newer RPI models will have wireless interfaces. Ignore wlan0 AND lo, focus on the MAC address for the eth0 interface.

Students will need to create a static DHCP mapping for the raspberry pi, using the MAC address they just recorded. It is absolutely vital that the bastion host's IP address never change. Unfortunately, I have no means on instructing you how to do this, as every network is different and every piece of network equipment has different instructions for configuring static DHCP mappings. I recommend referring to the documentation for the network equipment for guidance on how to do so. If the lab network is being configured on an enterprise or office network, students may also be required to work with network or system administrators in order to reserve a static DHCP allocation.

After configuring a static DHCP allocation, students should reboot their raspberry pi. Upon reboot, log in, run the `ip addr` command, and focus on the `inet` field. Confirm that the IP address displayed in that field matches the IP address of the static DHCP allocation. If the IP address matches, then the static DHCP mapping was successful.

```
raspberrypi login: pi
Password:
Linux raspberrypi 5.4.83+ #1379 Mon Dec 14 13:06:05 GMT 2020 armv6l

The programs included with the Debian GNU/Linux system are free software:
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ _

pi@raspberrypi:~$ passwd
Changing password for pi.
Current password:
New password:
Retype new password:
passwd: password updated successfully
pi@raspberrypi:~$ _

pi@raspberrypi:~$ sudo su -
root@raspberrypi:~# raspi-config
```

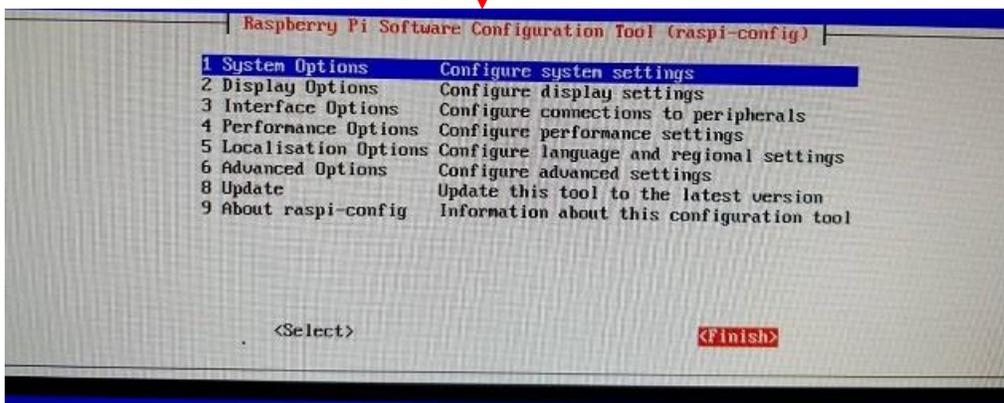
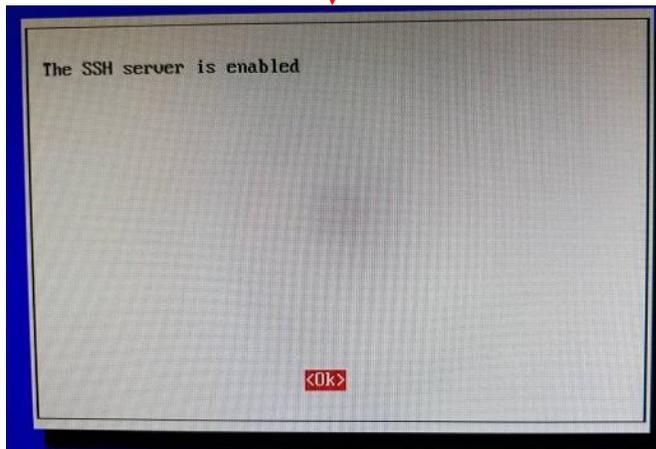
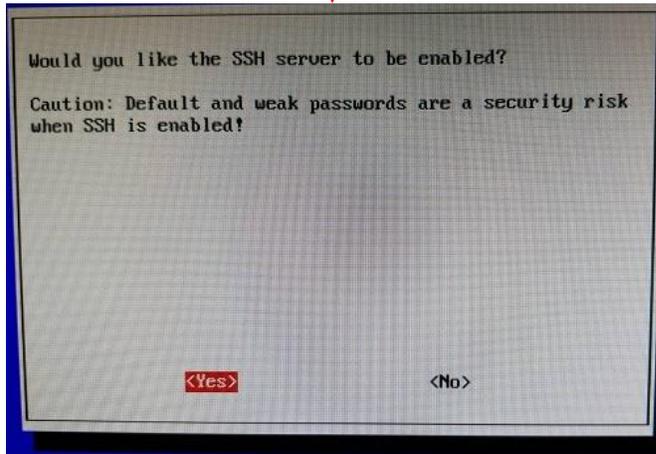
```
Raspberry Pi Software Configuration Tool (raspi-config)
1 System Options      Configure system settings
2 Display Options     Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options   Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config Information about this configuration tool
```

```
Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 UNC         Enable/disable graphical remote access using RealUNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins
```

Continued to fig. 16-17

16-16: Log in to the raspberry pi using the username and password combination, pi/raspberrypi. Run the passwd command and **immediately change the password of the pi user**. Afterwards, become the root user via sudo su -, and run raspi-config. From the main menu, select *Interface Options*, followed by *SSH*.

Continued from *fig. 16-16*



Continued to *fig. 16-18*

16-17: Students will be asked to confirm they want to enable the SSH service. Press enter to continue, get the confirmation that the service is enabled, then exit the raspi-config menu.

Continued from *fig. 16-17*

```
pi@raspberrypi:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether b8:27:eb:b3:77:a6 brd ff:ff:ff:ff:ff:ff
```

RPI	
IPv4 Address	
10.0.0.7	
IPv6 Address	Reserved IP
2601:408:502:c330::f853	
Local Link IPv6 Address	
fe80::4932:d1a4:254d:be93	
MAC Address	
B8:27:EB:B3:77:A6	
Comments	

```
pi@raspberrypi:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether b8:27:eb:b3:77:a6 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.7/24 brd 10.0.0.255 scope global dynamic noprefixroute eth0
```

16-18: Run the `ip addr` command and record the MAC address of the `eth0` interface from the `link/ether` field. Use that MAC address to create a static DHCP mapping on your local network. Afterwards, reboot the raspberry pi (either via `sudo init 6`, `reboot`, or unplug the board and plugging it back in). Upon reboot, run `ip addr` again, and confirm `eth0` was assigned its allocated IP address in the `inet` field.

Doing Things Manually, Raspberry Pi edition

Some students may need to manually set a static IP address, netmask, gateway, and DNS servers. How is this done? After all, Raspbian doesn't have a fancy Subiquity installer like Ubuntu does. You're going to need to get familiar with `/etc/dhcpd.conf`. Fortunately, the Raspberry Pi foundation has excellent documentation on the format for the file. Check out: <https://www.raspberrypi.org/documentation/configuration/tcpip/>

To modify the file, you'll need to become the root user (e.g., `sudo su -`), and use a command-line text editor – `vi`, `vim`, `nano`, `ed`, etc. Fortunately, the Raspbian maintainers included an example static IP configuration in the `dhcpd.conf` file that you can just modify to suit your needs:

```
# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

As an example, let's assume your raspberry pi is sitting on the `10.0.0.0/24` network. Your default gateway is `10.0.0.1`, and your preferred DNS servers are `8.8.8.8`, and `9.9.9.9`. You want the raspberry pi to have the IP address, `10.0.0.254`. Here is what that portion of the file would look like with that network configuration:

```
# Example static IP configuration:
interface eth0
static ip_address=10.0.0.254/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=10.0.0.1
static domain_name_servers=8.8.8.8 9.9.9.9
```

Notice the `interface eth0`, `static ip_address`, `static routers`, and `static domain_name_servers` lines are no longer commented out with a `"#"` symbol at the beginning of those lines? These are the lines you'll need to change. Once finished, the easiest way to reload the configuration changes you've just made to confirm they are working would be to reboot the raspberry pi, and run the `ip addr` and `ip route` commands to verify the proper IP address (the `inet` field of the `eth0` interface), and default gateway were applied successfully.

16.3.3 Configuring Static Routes on the Bastion Host

A brief introduction to network routing, and static routes was covered in Chapter 15, [section 15.1](#) (pp. 727-731). To make a long story short, in order to communicate with the lab virtual machines, the bastion host needs to have network routes to know where to send the network traffic in order to reach the destination. However, different Linux distributions used different network management software to do things.

In the way back when™, Most Linux distributions used the `/etc/network/interfaces` configuration file. Then Systemd happened, and `systemd-networkd` was supposed to be a single, unified way to handle network configuration. But some distributions decided they didn't like that, and used their own solutions, instead. Raspbian uses `dhcpcd`, while Ubuntu uses `netplan`. In the subsections that follow, students will learn how to create persistent static routes on Ubuntu using `netplan`, and on Raspbian using `dhcpcd`.

Note: Students will need to modify files on either their Ubuntu bastion host VM, or raspberry pi. In order to do so, students will need to use a command line text editor of some sort (e.g., `vi/vim`, `nano`, `ed`, etc.) to make the necessary modifications. If you're not yet comfortable with Unix/Linux command line text editors, I recommend taking some time to get a little more comfortable. Make use of the training resources listed in chapter 2, [section 2.3](#), pp. 38-39

16.3.3.1 Persistent Static Routes on Ubuntu, using `netplan`

Log on the Ubuntu bastion host VM, and use the command:

```
sudo su -
```

to become `root`. The file students will need to modify is `/etc/netplan/00-installer-config.yaml`. This file is, essentially a configuration file that informs the operating system on how networking is configured on Ubuntu. **Before students do anything to this file, make an emergency backup copy by running the following command:**

```
cp /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.bak
```

Do not skip this step, because if the file is misconfigured in any way, more than likely it will result in there being no network connectivity on the virtual machine when it is rebooted. In which case, the best solution would be to restore the virtual machine from a snapshot, or using the `cp` command to copy the backup file over the misconfigured file.

```
network:
  ethernets:
    ens160:
      dhcp4: true
  version: 2
```

Using whatever text editor students are most comfortable with, students will need to modify the file to make it look like this:

```
network:
  ethernets:
    ens160:
      dhcp4: true
      routes:
        - to: 172.16.1.0/24
          via: 10.0.0.26
        - to: 172.16.2.0/24
          via: 10.0.0.26
  version: 2
```

Notice how almost all of the fields are nested, and indented exactly two spaces after the previous field. For example, the `network` field has no indentation, but the `ethernets` field below it on the next line is indented exactly two spaces.

Note the field labeled `ens160`. This is the name of the network interface on my Ubuntu virtual machine, and may possibly differ for students following along at home. Use the command `ip -br addr` to display the name of the network interface on the bastion virtual machine, and compare that to the contents of the `00-installer-config.yaml` file.

Next, take notice of the `routes` section **in bold**. Every route added to the netplan configuration file consists of both a `to` field, that tells netplan the network students want a route to, and the `via` field, telling netplan which IP address to forward packets to in order to reach that destination network. The configuration file above for my bastion virtual machine says "On startup, generate two static routes. One route to the the 172.16.1.0/24 network, and another route to the 172.16.2.0/24 network. For any traffic destined to these two networks, send that traffic to the IP address 10.0.0.26." That IP address is the IP address of the WAN interface of the pfSense VM on my local network.

If students are using a custom network configuration for their lab environment (see Chapter 13, section 13.6.4.2. Specifically, the sidebar conversation, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range*), then make sure to alter the network destinations in both of the `to` fields accordingly. Likewise, I don't expect your pfSense VM's WAN interface to have the IP address 10.0.0.26, so the `via` field for both of those entries will need to be altered as well.

After students have finished modifying the file, save it, and run the command:

```
netplan try
```

If everything was modified successfully, students will get the following output:

```
Warning: Stopping systemd-network.service, but it can still be activated by:
  systemd-networkd.socket
Do you want to keep these settings?
```

Press ENTER before the timeout to accept the new configuration

Pressing enter will reload the network configuration, and add the necessary static routes. Run the `reboot` command to restart the virtual machine. Then, upon reboot, run the following commands to confirm the bastion VM has its network configuration, and that the static routes students configured are present in the routing table:

```
ip -br addr
ip route
```

What happens when `netplan try` finds a problem with the configuration file? Here is what happens when I specify the `routes` field, but I don't have any `to` or `from` fields nested underneath it to tell `netplan` what routes to add:

```
/etc/netplan/00-installer-config.yaml:6:14: Error in network definition:
expected sequence
  routes:
    ^
```

An error occurred: the configuration could not be generated

Reverting.

```
Warning: Stopping systemd-networkd.service, but it can still be activated by:
  systemd-networkd.socket
```

`netplan try` will tell students the name of the file it found the problem with (`/etc/netplan/00-installer-config.yaml`), the line number (6) and where on that line (column 14) it encountered the problem. Students can then modify the configuration file, and run `netplan try` again to see if the problem is fixed, or if there are other problems in the file that need to be corrected. Keep doing this until the `netplan try` stops complaining about the file, or restore the original `00-installer-config.yaml` from backup, and start over from scratch by running the following command (as the `root` user):

```
cp /etc/netplan/00-installer-config.yaml.bak /etc/netplan/00-installer-config.yaml
```

With the original file restored, students can start over from scratch, and try editing the file again.

```

ayy@bastion:~$ sudo su -
[sudo] password for ayy:
root@bastion:~# cp /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.bak
root@bastion:~# vi /etc/netplan/00-installer-config.yaml
root@bastion:~# cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      dhcp4: true
      routes:
        - to: 172.16.1.0/24
          via: 10.0.0.26
        - to: 172.16.2.0/24
          via: 10.0.0.26
  version: 2
root@bastion:~# _
root@bastion:~# netplan try
Warning: Stopping systemd-networkd.service, but it can still be activated by:
systemd-networkd.socket
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 119 seconds
Configuration accepted.
root@bastion:~# reboot
ayy@bastion:~$ ip -br addr
lo                UNKNOWN          127.0.0.1/8 ::1/128
ens160            UP                10.0.0.162/24 2601:408:502:c330::1b46/128
:fe93:8624/64 fe80::20c:29ff:fe93:8624/64
ayy@bastion:~$ ip route
default via 10.0.0.1 dev ens160 proto dhcp src 10.0.0.162 metric 100
10.0.0.0/24 dev ens160 proto kernel scope link src 10.0.0.162
10.0.0.1 dev ens160 proto dhcp scope link src 10.0.0.162 metric 100
172.16.1.0/24 via 10.0.0.26 dev ens160 proto static onlink
172.16.2.0/24 via 10.0.0.26 dev ens160 proto static onlink

```

16-19: To modify Ubuntu's netplan configuration files, become the root user, and begin by using the `cp` command to make a backup of `/etc/netplan/00-installer-config.yaml`. Afterwards, using whichever text editor students are most familiar with, add the routes, `- to`, and `via` sections to the configuration file. Once finished, save the file, then run `netplan try`. After the configuration changes have been validated, reboot the bastion VM. Upon reboot, run `ip -br addr`, and `ip route` to confirm the bastion VM has its assigned IP address, and that the static route entries to the lab network show up in the routing table.

What if I set a static IP address instead?

Some of you might have configured a static IP address, netmask, gateway, DNS servers, etc. for their bastion virtual machine during the initial operating system install for whatever reason. Your `00-installer-config.yaml` file will probably look something like this as a result:

```
network:
  ethernets:
    ens160:
      addresses:
        - 10.0.0.250/24
      gateway4: 10.0.0.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
        search: []
  version: 2
```

Of course, your `00-installer-config.yaml` file will vary based on the IP address, default gateway, DNS servers and suffixes you've configured. Become the root user, and run:

```
cp /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.bak
```

Then make the following changes:

```
network:
  ethernets:
    ens160:
      addresses:
        - 10.0.0.250/24
      gateway4: 10.0.0.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
        search: []
      routes:
        - to: 172.16.1.0/24
          via: 10.0.0.26
        - to: 172.16.2.0/24
          via: 10.0.0.26
  version: 2
```

Save the changes you've made, run `netplan try` to validate the changes, then reboot the VM. Upon reboot, log back in and run `ip -br addr`, and `ip route` to confirm the IP address of the network interface is intact, and that the static routes to the lab network were added to the routing table.

```

ayy@avenged:~$ sudo su -
root@avenged:~# cp /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.bak
root@avenged:~# vi /etc/netplan/00-installer-config.yaml
root@avenged:~# cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    ens160:
      addresses:
        - 10.0.0.250/24
      gateway4: 10.0.0.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
        search: []
      routes:
        - to: 172.16.1.0/24
          via: 10.0.0.26
        - to: 172.16.2.0/24
          via: 10.0.0.26
  version: 2
root@avenged:~# netplan try
Warning: Stopping systemd-networkd.service, but it can still be activated by:
systemd-networkd.socket
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 62 seconds
Configuration accepted.
root@avenged:~# reboot_
ayy@avenged:~$ ip -br addr
lo                UNKNOWN          127.0.0.1/8 ::1/128
ens160            UP                10.0.0.250/24 2601:408:502:c330::a615/128
:fe5b:b2e3/64 fe80::20c:29ff:fe5b:b2e3/64
ayy@avenged:~$ ip route
default via 10.0.0.1 dev ens160 proto static
10.0.0.0/24 dev ens160 proto kernel scope link src 10.0.0.250
172.16.1.0/24 via 10.0.0.26 dev ens160 proto static
172.16.2.0/24 via 10.0.0.26 dev ens160 proto static

```

16-20: The process of modifying the 00-installer-config.yaml file is more or less identical for students with static IP address configurations. There will be a few more additional fields to worry about, but just append the routes, to, and via fields as displayed above. Afterwards, save the changes, run netplan try to validate the configuration file and ensure there are no errors, then reboot the virtual machine. Upon reboot, log back in, run ip -br addr and ip route to verify the interface's IP address is the same, and that the static routes were successfully added to the routing table.

16.3.3.2 Persistent Static Routes on Raspbian, using dhcpcd

Log on to your raspberry pi, and run the command `sudo su -` to become the root user. Next, using their preferred text editor (or output redirection, if preferred), students will need to create the file, `/lib/dhcpcd/dhcpcd-hooks/40-route`. Fortunately, this is a very simple file in which students only need to add the following two lines:

```
ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26
```

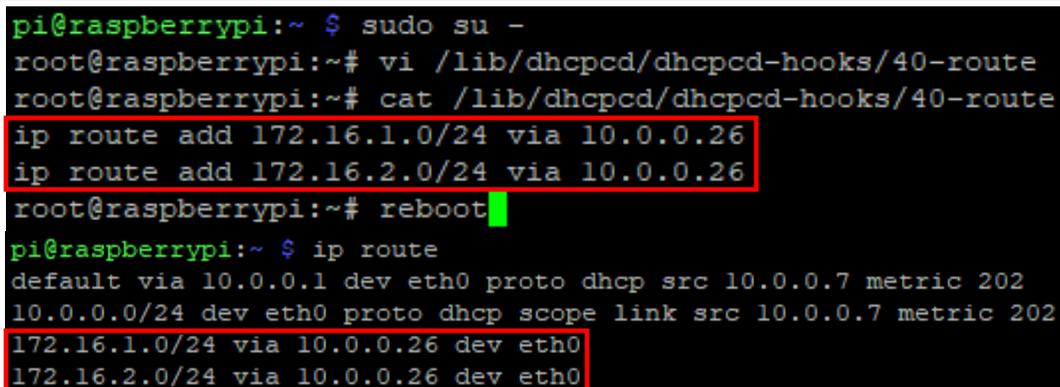
If students are using a custom network configuration for their lab environment (see Chapter 13, section 13.6.4.2. Specifically, the sidebar conversation, *What do I do if my home or office network is using 172.16.1.0/24, 172.16.2.0/24 or the entire 172.16.0.0/12 RFC1918 range*), then make sure to alter the network destinations in both lines accordingly. Likewise, I don't expect your pfSense VM's WAN interface to have the IP address 10.0.0.26, so the `via` field for both of those entries will need to be altered as well.

After students have finished modifying the file, save it, and run the `reboot` command. Once the system has finished rebooting log back in, and run the `ip route` command to confirm that the static routes have been successfully added to the routing table.

Note: If you're still not comfortable with text editors, you can use output redirection to add the two `ip route` commands, and create the `/lib/dhcpcd/dhcpcd-hooks/40-route` file. Run the following commands:

```
echo "ip route add 172.16.1.0/24 via 10.0.0.26" > /lib/dhcpcd/dhcpcd-hooks/40-route
echo "ip route add 172.16.2.0/24 via 10.0.0.26" >> /lib/dhcpcd/dhcpcd-hooks/40-route
```

As demonstrated in fig. 16-21, run the `cat` command to confirm the commands got copied to the file correctly, then `reboot` the raspberry pi and run `ip route` to verify the static routes were added successfully.



```
pi@raspberrypi:~ $ sudo su -
root@raspberrypi:~# vi /lib/dhcpcd/dhcpcd-hooks/40-route
root@raspberrypi:~# cat /lib/dhcpcd/dhcpcd-hooks/40-route
ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26
root@raspberrypi:~# reboot
pi@raspberrypi:~ $ ip route
default via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.7 metric 202
10.0.0.0/24 dev eth0 proto dhcp scope link src 10.0.0.7 metric 202
172.16.1.0/24 via 10.0.0.26 dev eth0
172.16.2.0/24 via 10.0.0.26 dev eth0
```

16-21: Log in to the raspberry pi, and become the root user. Students will need to create the file `/lib/dhcpcd/dhcpcd-hooks/40-route`. This file should contain two lines with an `ip route add` command on each line, Using the illustration above for guidance. When finished, save the file, and reboot the raspberry pi. Upon reboot, log back in, and run the command `ip route` to confirm the new static routes have been added to the routing table successfully.

A Third Option: rc.local

If you're having problems with `netplan` or `dhcpcd`, there's another method that can be used to add static routes to your bastion host on boot: `rc.local`. To make a very long story short, `rc.local` is a legacy Linux component. It's a script that is the last thing that gets run automatically when a system is booted. All students need to do is make a small shell script that contains the necessary `ip route add` commands, and the system will automatically add those routes on boot. However, there's one slight problem: Since `rc.local` is considered a "legacy" function on modern Linux distributions, this means that support could be dropped at any time. At the time of mine writing this (Early 2021), both Ubuntu Server 20.04 and Raspbian both support the use of `/etc/rc.local`. The file does not exist by default on Ubuntu Server 20.04, but does exist on Raspbian (and has a small macro configured).

`/etc/rc.local` on Ubuntu Server 20.04

Since the file `/etc/rc.local` doesn't exist on Ubuntu, students will have to create it, using a command line text editor of their choice. Log in to your bastion VM and become the root user (using `sudo su -`). Run the command `vi /etc/rc.local` and write the following lines into the file:

```
#!/usr/bin/env bash
ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26
exit 0
```

As always, if students have a different network configuration for their lab environment or a different IP address assigned for the WAN interface of the pfSense VM, substitute the networks and IP addresses as necessary. When finished, save the file, and run:

```
chmod 700 /etc/rc.local
reboot
```

When the system is finished rebooting, log in and run the command `ip route` to confirm the routing table was updated successfully.

```

root@bastion:~# vi /etc/rc.local
root@bastion:~# cat /etc/rc.local
#!/usr/bin/env bash
ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26
exit 0
root@bastion:~# chmod 700 /etc/rc.local
root@bastion:~# reboot
aay@bastion:~$ ip route
default via 10.0.0.1 dev ens160 proto dhcp src 10.0.0.162 metric 100
10.0.0.0/24 dev ens160 proto kernel scope link src 10.0.0.162
10.0.0.1 dev ens160 proto dhcp scope link src 10.0.0.162 metric 100
172.16.1.0/24 via 10.0.0.26 dev ens160
172.16.2.0/24 via 10.0.0.26 dev ens160

```

16-22: Students will have to create `/etc/rc.local` in the correct format with the necessary commands to update the routing table for their home network and lab environment. Afterwards, the file needs to have execute permissions added via `chmod 700 /etc/rc.local`. Afterwards, reboot the system, log back in and run `ip route` to verify that the static routes have been added to the routing table.

`/etc/rc.local` on Raspbian

On Raspbian, `/etc/rc.local` already exists, and ready to go. By default, the file has a small macro that displays the IP address of the system on boot:

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
exit 0

```

To modify the existing `/etc/rc.local` file to generate static routes for the lab environment, log in to your raspberry pi, and become the root user (via `sudo su -`). Using your favorite command-line text editor, add the lines **in bold** to the `/etc/rc.local` file:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26

exit 0
```

As always, if students have a different network configuration for their lab environment or a different IP address assigned for the WAN interface of the pfSense VM, substitute the networks and IP addresses as necessary. When finished, save the file and reboot the raspberry pi. When the system is finished rebooting, log in and run the command `ip route` to confirm that the routing table was updated successfully.

```

pi@raspberrypi:~ $ sudo su -
root@raspberrypi:~# vi /etc/rc.local
root@raspberrypi:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

ip route add 172.16.1.0/24 via 10.0.0.26
ip route add 172.16.2.0/24 via 10.0.0.26

exit 0
root@raspberrypi:~# reboot
pi@raspberrypi:~ $ ip route
default via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.7 metric 202
10.0.0.0/24 dev eth0 proto dhcp scope link src 10.0.0.7 metric 202
172.16.1.0/24 via 10.0.0.26 dev eth0
172.16.2.0/24 via 10.0.0.26 dev eth0

```

16-23: /etc/rc.local already exists on Raspbian (at least as of Raspbian 10), so all students need to do is modify the existing file, add in the `ip route add` commands for static routes to the lab network segments, save their changes to the file, then reboot, and confirm the routes were successfully added to the routing table via the `ip route` command.

16.3.4 Configuring the pfSense Firewall

As mentioned in Chapter 15, network routes are one component in network configuration. They tell a computer *how* to reach a particular destination. Another part of network configuration of our bastion host, is whether or not it is *allowed* to access systems on those networks. Students will need to modify some of the firewall rules on the WAN interface of the pfSense VM, using the pfSense webConfigurator.

Students will need to have access to the pfSense webConfigurator in order to make these changes. Back in chapter 14, we established access to the webConfigurator on the pfSense VM's WAN interface, specifically in [section 14.4.5.1](#) (pp. 711-712). In order to do this initially, I told students to use the IP address of their workstation for the access rules. If the IP address of the workstation has changed, students will need to regain access to the webconfigurator. Fortunately, I also covered how to do this in [section 14.4.5.5](#), specifically, in the sidebar discussion, [Have You Looked in the Mailbox](#) (pp. 721-724).

Log in to the pfSense webConfigurator, and navigate to *Firewall > Aliases*. We're going to create an IP address alias, so ensure that the *IP* tab is selected. Students will know that they're in the correct tab because the *RFC_1918_Addresses* aliases created in chapter 14 will be displayed. Click the *Add* button in the lower right corner.

On the next page, labeled *Edit* enter the following information in the fields under the *Properties* section:

Name	Bastion_Hosts
Description	A host, or group of hosts to be used for accessing the lab networks.
Type	Host(s)

Next, under the *Host(s)* section, enter the IP addresses of hosts you would like to have remote access to the lab environment. Most students will be entering only a single IP address – either the IP address of their bastion host virtual machine, or the IP address of the raspberry pi (or other physical system) we configured together in this chapter. However, some students may wish to enable multiple bastion hosts, or workstations access to the lab environment. If you'd like to allow more than one IP address access to the lab networks hosted on the bare-metal hypervisor, click the *Add Host* button to create additional input boxes for more IP addresses. I also recommend a robust description for every IP address added to this alias. For example, this is what I chose to enter:

IP Address	Description
10.0.0.6	Static IP address of my Workstation
10.0.0.7	Static IP address of my Raspberry Pi
10.0.0.162	Static IP address of my Ubuntu bastion host VM

Next, click the *Save* button under the *Host(s)* section of the page, then back the IP tab of the firewall aliases page, click the *Apply Changes* button to create the new alias.

The next step of this plan involves modifying several firewall rules on the WAN interface. Navigate to *Firewall > Rules* and ensure that the *WAN* tab is selected. Recall in Chapter 14, [section 14.4.5.1](#) (pp. 711-712), how we created a whole host of rules to access various services in the lab environment. I instructed students to configure their workstation's IP address as the source IP address for these rules. For example, the source IP address I used for my lab network was 10.0.0.6. Edit all of these firewall rules (Click the Pencil icon under the Actions column), and change the *single host or alias* input box in the *Source* section from the IP address specified in chapter 14, to the *Bastion_Hosts* alias. Afterwards, click *Save*. There should be five firewall rules that need to be modified in total. Once all five rules have been successfully modified, Click *Apply Changes* to reload the *WAN* interface firewall policy.

Firewall / Aliases / IP [List] [?]

IP | Ports | URLs | All

Firewall Aliases IP

Name	Values	Description	Actions
RFC_1918_Addresses	192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8	Network alias for all IPv4 RFC1918 networks	[Edit] [Copy] [Delete]

+ Add [Import]



Firewall / Aliases / Edit

Properties

Name
The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".

Description
A description may be entered here for administrative reference (not parsed).

Type

Host(s)

Hint Enter as many hosts as desired. Hosts must be specified by their IP address or fully qualified domain name (FQDN) re-resolved and updated. If multiple IPs are returned by a DNS query, all are used. An IP range such as 192.168.1 as 192.168.1.16/28 may also be entered and a list of individual IP addresses will be generated.

IP or FQDN	
<input type="text" value="10.0.0.6"/>	<input type="text" value="Static IP address of my Workstation"/>
<input type="text" value="10.0.0.7"/>	<input type="text" value="Static IP address of my Raspberry Pi"/>
<input type="text" value="10.0.0.162"/>	<input type="text" value="Static IP address of my Ubuntu bastion host VM"/>

[Save] + Add Host



The alias list has been changed.
The changes must be applied for them to take effect. [Apply Changes]

IP | Ports | URLs | All

Firewall Aliases IP

Name	Values	Description	Actions
Bastion_Hosts	10.0.0.6, 10.0.0.7, 10.0.0.162	A host, or group of hosts to be used for accessing the lab networks.	[Edit] [Copy] [Delete]

16-24: Navigate to *Firewall* > *Aliases* and under the IP tab, click *Add*. Fill out the *Bastion_Hosts* alias illustrated above (substituting the IP address for your physical or virtual bastion host as necessary, and/or creating additional bastion hosts), then click *Save*, Followed by *Apply Changes* to create the new alias.

Firewall / Rules / WAN

Floating WAN LAN OPT1

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 2 / 1.19 MiB	IPv4 TCP	10.0.0.6	*	10.0.0.26	443 (HTTPS)	*	none		Better Anti-Lockout rule	
✓ 0 / 0 B	IPv4 TCP	10.0.0.6	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.6	*	172.16.1.3	8000	*	none		Allow splunk Web access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.6	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to IPS VM	
✓ 0 / 0 B	IPv4 TCP	10.0.0.6	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to Kali VM	
✗ 0 / 17.08 MiB	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any WAN	

Source

Source Invert match

Single host or alias

Bastion_Hosts

Save

The firewall rule configuration has been changed.
The changes must be applied for them to take effect.

Apply Changes

Floating WAN LAN OPT1

Rules (Drag to Change Order)

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓ 1 / 1.37 MiB	IPv4 TCP	Bastion_Hosts	*	10.0.0.26	443 (HTTPS)	*	none		Better Anti-Lockout rule	
✓ 0 / 0 B	IPv4 TCP	Bastion_Hosts	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	Bastion_Hosts	*	172.16.1.3	8000	*	none		Allow splunk Web access to SIEM VM	
✓ 0 / 0 B	IPv4 TCP	Bastion_Hosts	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to IPS VM	
✓ 0 / 0 B	IPv4 TCP	Bastion_Hosts	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to Kali VM	
✗ 0 / 17.70 MiB	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any WAN	

16-25: with the Bastion_Hosts alias created, Navigate to *Firewall > Rules* and ensure the *WAN* tab is selected. For all five allow rules created in chapter 14, click the pencil icon under the *Actions* column to edit the firewall rules individually. Students will change the source from the IP address of their management workstation to the Bastion_Hosts alias. After all five rules have been modified, be sure to click *Apply Changes* to update the WAN interface's firewall rules.

16.4 SSH, SSH Tunnels, and You

So far, in this chapter we've covered how to:

- Create a (physical or virtual) bastion host
- Automatically generate static routes for bastion hosts to know how to reach the lab network segments
- Reconfigure firewall rules to allow bastion hosts access to the lab virtual machines

All of that effort has been in preparation for what students will need to accomplish in the remaining sections of this chapter. Students will learn about how to access their bastion host using the SSH protocol. Students will also learn about TCP forwarding over SSH, and use that knowledge to create forward tunnels that allow them to log in to the virtual machines in their lab environment through the SSH connection established on their bastion host. Students will also learn how to create Dynamic SSH tunnels and use them in combination the FoxyProxy add-on for Mozilla Firefox or Google Chrome in order to treat their bastion host as an HTTP/HTTPS proxy to access web applications in their lab environment as well. Once students have successfully established SSH connectivity to their bastion host, and confirmed that their tunneled SSH and HTTP/HTTPS connections to the lab environment are working correctly, students will then have the option to configure key-based authentication for the bastion host and lab virtual machines, and/or enable SSH access to lab virtual machines as the root user.

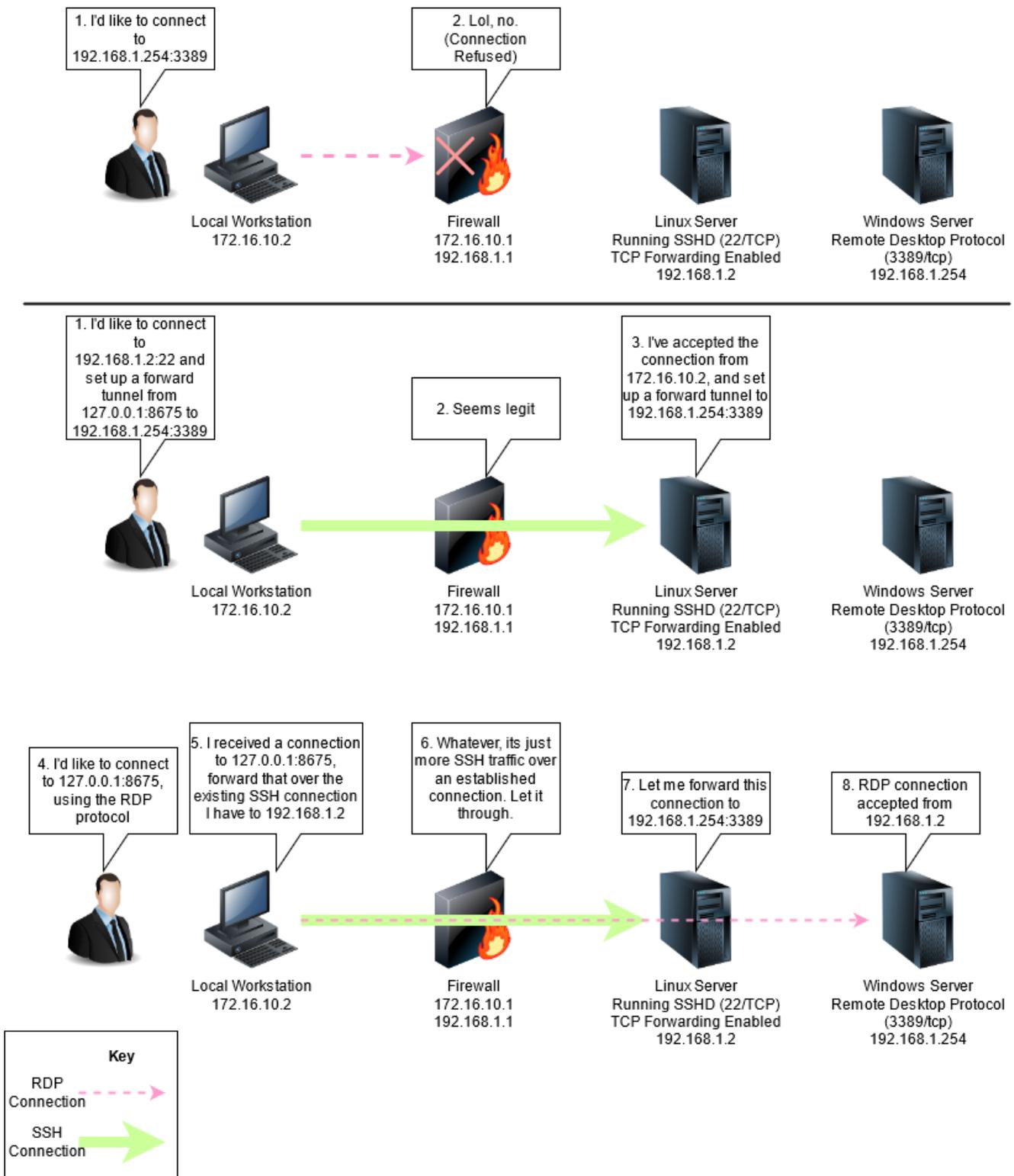
A lot of the content that will be covered in these remaining sections overlaps very heavily with the content covered in chapter 15. Therefore, I will be referring students back to specific sections in Chapter 15 for guidance on how to perform particular tasks that I already covered for the hosted hypervisor users. However, detailed illustrations that describe how to perform all of the actions required for Windows, Linux and/or MacOS users will still be provided.

16.4.1 SSH Tunneling Explained

SSH is an encrypted remote access protocol that can be used to control a system over a network connection. SSH has evolved over the years, gaining stronger encryption, better security, and more capabilities. One of its more interesting functions is something called *TCP Forwarding*. In a nutshell, if the server a user is connecting to allows it, TCP forwarding allows a client to treat the server sort of like a proxy, through *local*, *remote*, and *dynamic* TCP forwarding. Let's talk more about the different types of TCP forwarding.

Note: TCP forwarding is more commonly known as *SSH tunneling*. Local TCP forwarding is often referred to as *forward tunneling*, while remote TCP forwarding is referred to as *reverse tunneling*. Dynamic TCP forwarding is known as a *dynamic tunnel*. This is the terminology I will be using for the rest of the chapter.

16.4.1.1 Forward Tunnels, Illustrated



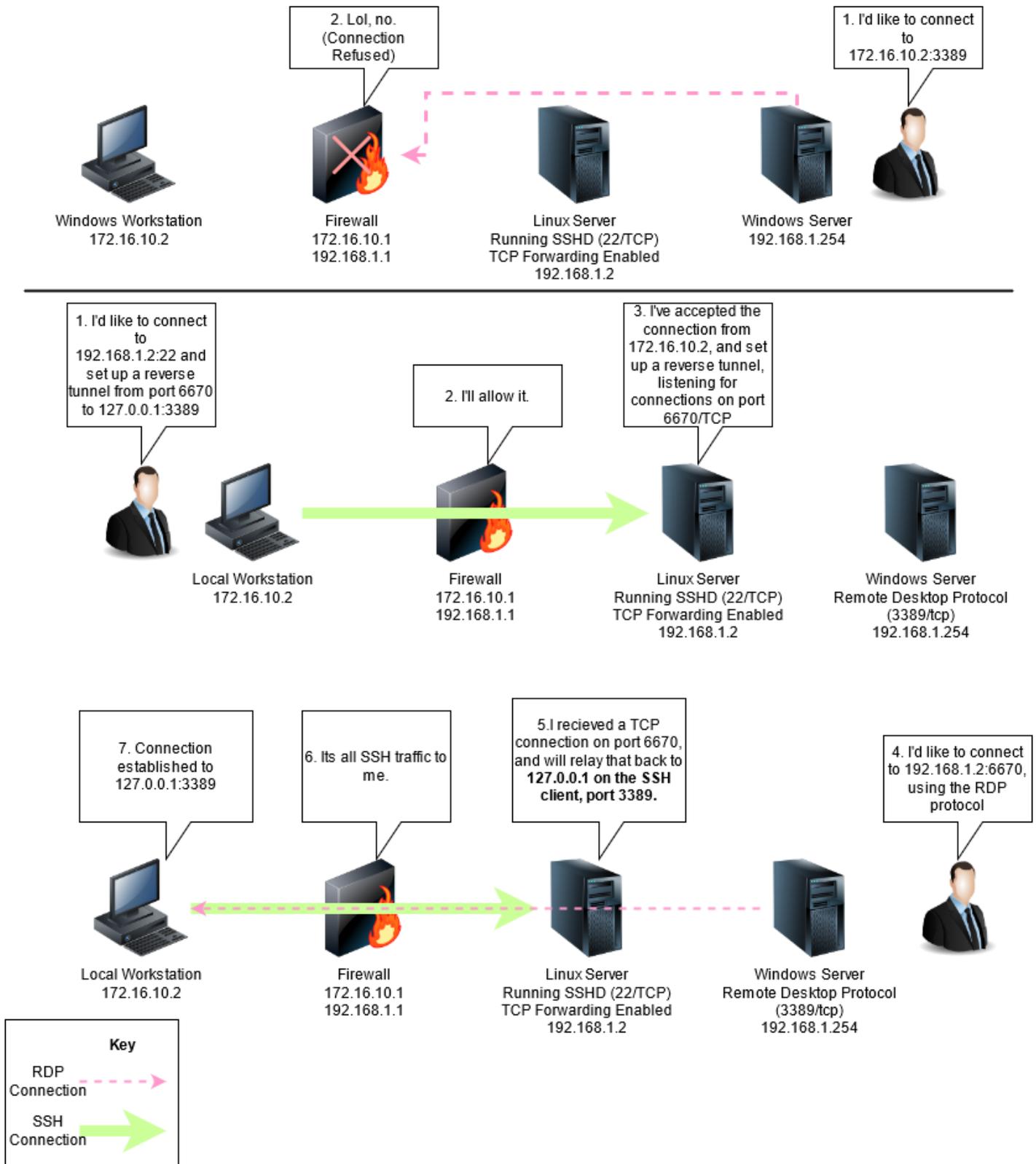
16-26: This diagram illustrates an example on how forward tunnels work. The Linux server at 192.168.1.2 that the user at 172.16.10.2 is allowed to connect to over SSH is used as a relay. This allows the user to bypass the firewall, and make the RDP connection to 192.168.1.254 look like its coming from 192.168.1.2.

Figure 16-26 demonstrates an example on how forward tunnels work. In the example, the user at 172.16.10.2 wants to connect to a Windows system at 192.168.1.254, using the Remote Desktop Protocol (port 3389/TCP). The firewall between them denies the connection. However, the user is able to establish an SSH connection to a Linux server at 192.168.1.2 (port 22/TCP). The SSH server has TCP forwarding enabled, so the user creates an SSH connection and creates a forward tunnel.

The way forward tunnels work is that the SSH client establishes a connection to the server, and through a special configuration option, creates a network service on the SSH client that listens for connections on an arbitrary TCP port that the user can choose. The example above uses port 8675/TCP. Any time a connection is received on the configured port, that connection is relayed over the SSH connection and to the destination configured – in this case, 192.168.1.254 on port 3389/TCP. Most of the time this network service listens on the IP address 127.0.0.1, the local system's loopback IP address. This prevents other systems on the same network as the SSH client from using the forward tunnel. However, there are configuration options that will accept connections on the configured TCP port from all interfaces with an IP address, allowing other systems on the network to use the configured forward tunnel as well. Since the forwarded TCP connection looks like it's coming from 192.168.1.2, this allows the user in our example to bypass the firewall, and access the Windows system.

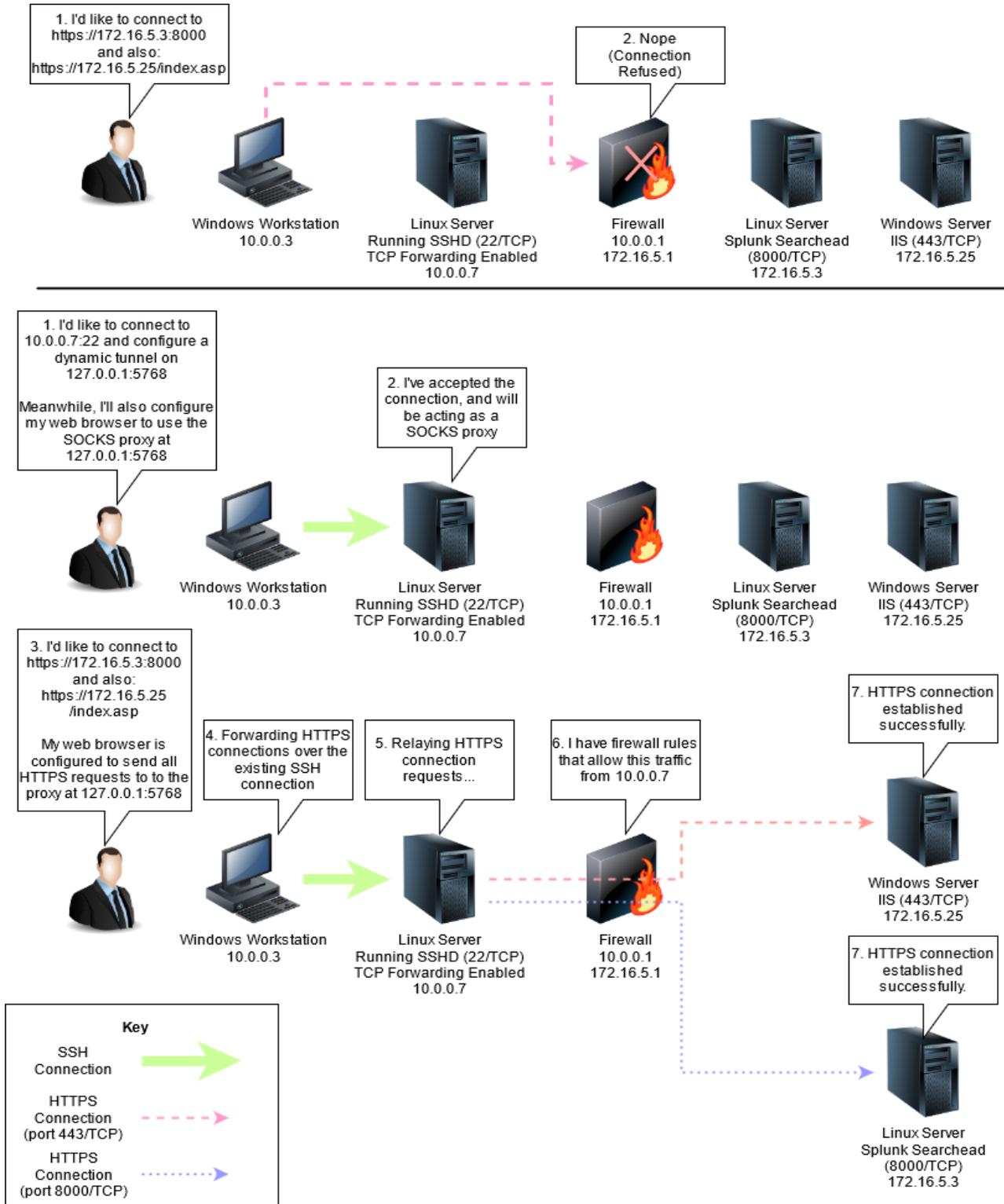
16.4.1.2 Reverse Tunnels, Illustrated

To better explain reverse tunnels, let's take the scenario above and change things a little bit. Let's say a user at the 192.168.1.254 system wants to access the remote desktop service on the 172.16.10.2 system on port 3389/TCP, but once again, the firewall is blocking this connectivity. The user at 172.16.10.2 could create an SSH connection to 192.168.1.2 again with special configuration options, a network listener (on port 6670/TCP) would be opened on the system 192.168.1.2. Now, the user at 192.168.1.254 would tell their RDP client to connect to 192.168.1.2:6670, and the connection would get forwarded back across the SSH connection to 172.16.10.2. Take a look at *fig. 16-27* below for an illustration to make this a little less confusing.



16-27: Reverse tunnels can be a little bit confusing to understand at first. With a reverse tunnel, a listener is opened on the remote side of the network connection (at the SSH server the client is connecting to). When this listener receives a connection, it is relayed back across the SSH connection towards the SSH client's network. If students are familiar with reverse HTTP proxies (e.g., NGINX, etc.) reverse tunnels are very similar.

16.4.1.3 Dynamic Tunnels, Illustrated



16-28: Dynamic tunneling allows users to treat the server side of an SSH connection like a proxy. While this example demonstrates using it as an HTTP/HTTPS proxy, technically the dynamic tunnel could be used to proxy traffic that is compatible with SOCKSv4.

When users configure a dynamic tunnel, it creates a listener on the SSH client system, similar to forward tunnels, however dynamic tunnels do not specify a destination IP address and port combination. This allows the dynamic tunnel to operate identically to something called a SOCKS (shorthand for SOCKeT\$) proxy. In the example above, this is demonstrated by establishing the SSH connection to 10.0.0.7 with a dynamic tunnel configured to listen on 127.0.0.1:5768.

After establishing the SSH session, the user at 10.0.0.3 configures their web browser to use a proxy server located at 127.0.0.1:5768. Notice how in the example there are multiple destinations the user at 10.0.0.3 wants to visit in the 172.16.5.0/24 network? SOCKS proxies/dynamic tunnels don't need a specific destination defined, they forward traffic to the destinations requested as transparently as possible.

Overlapped Defense

SSH Tunneling (along with many other methods) is often used by offensive security practitioners to mask where attacks are coming from and gain further access into target networks, as a part of a method called *pivoting*. As demonstrated by the diagrams above, all the network firewall sees in these scenarios is SSH traffic. So, if SSH traffic is allowed, and TCP Forwarding is enabled, the network firewall doesn't know any better, and won't be able to help. This is why defense in depth is such an important security concept, because no single security control is without weakness. In the case of SSH tunneling, other mitigations are necessary to prevent pivoting, such as robust Endpoint Detection and Response (EDR) software, robust audit logging, and enabling endpoint firewalls (such as Windows Firewall or Linux IPtables, etc.) are all mitigations that can help to cover the gaps that network firewalls miss.

More about SSH

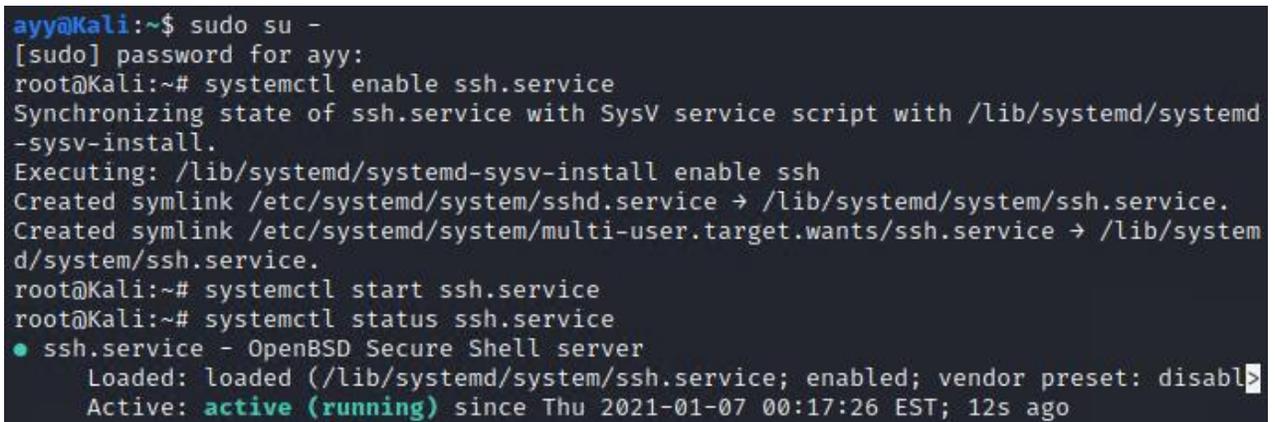
SSH is a very old protocol with a lot of other functionality buried deep within. You could use it for years, and still only utilize a fraction of its full power. If you're interested in learning more, O'Reilly books has a gigantic tome about SSH that features a snail on the cover – *SSH, The Secure Shell: The Definitive Guide*. I promise you'll learn more about SSH than you ever really wanted to.

16.4.2 Enabling the SSH service on the Kali Linux VM

By default, the Kali Linux VM ships with the SSH service disabled. While not strictly necessary, enabling SSH access to the kali VM is both very convenient, and recommended. Fortunately, the instructions on how to do this are both easy, and were already covered in Chapter 15, [section 15.1.4](#) (p. 742). But to make a long story short, log in to the Kali VM using the virtual console on the ESXi web interface. Open the terminal application and run the following commands:

```
sudo su -
systemctl enable ssh.service
systemctl start ssh.service
```

Afterwards, students may log out of the Kali VM's virtual console.



```
ayy@Kali:~$ sudo su -
[sudo] password for ayy:
root@Kali:~# systemctl enable ssh.service
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd
-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/system
d/system/ssh.service.
root@Kali:~# systemctl start ssh.service
root@Kali:~# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabl
   Active: active (running) since Thu 2021-01-07 00:17:26 EST; 12s ago
```

16-29: In order to be able to connect to Kali over SSH, students will need to open a console session to the Kali VM on the ESXi web interface, log in, open a terminal session then enter the commands specified above to re-enable then start the ssh service. **After running these commands, take a new snapshot of the Kali VM.**

16.5 Establishing SSH Connectivity to the Bastion Host and Lab VMs (Windows)

In this section, I will instruct students on how to configure SSH connections to their bare-metal hypervisor lab on Windows using mRemoteNG. We will first establish SSH connectivity to the bastion host, then configure forward tunnels to the SIEM, IPS, and Kali virtual machines, as well as a dynamic tunnel for later use. After connectivity to the bastion host and lab virtual machines has been established, students will be guided through the process on how to generate an SSH public/private key pair, as well as how to enable key-based authentication to their lab virtual machines, and bastion host. **MacOS and Linux users should skip ahead to section 16.6 on page 919.**

16.5.1 Connecting to the Bastion Host with mRemoteNG

To begin, open the mRemoteNG application, and under the *Connections* pane, click the *New Connection* icon. This creates a new mRemoteNG connection profile. A whole host of new fields will appear under the *Config* pane in the lower left corner. Under the *Display* section, in the *Name* field, enter `bastion_host`. Under the *Protocol* section, in the *Protocol* field, select *SSH version 2* from the drop-down. Finally, in the *Connection* section, students should enter the IP address of their bastion host in the *Hostname/IP* field, and the username they supplied in the *Username* field. Leave the rest of the fields their default values. For example, this is the data I entered to connect to my raspberry pi bastion host:

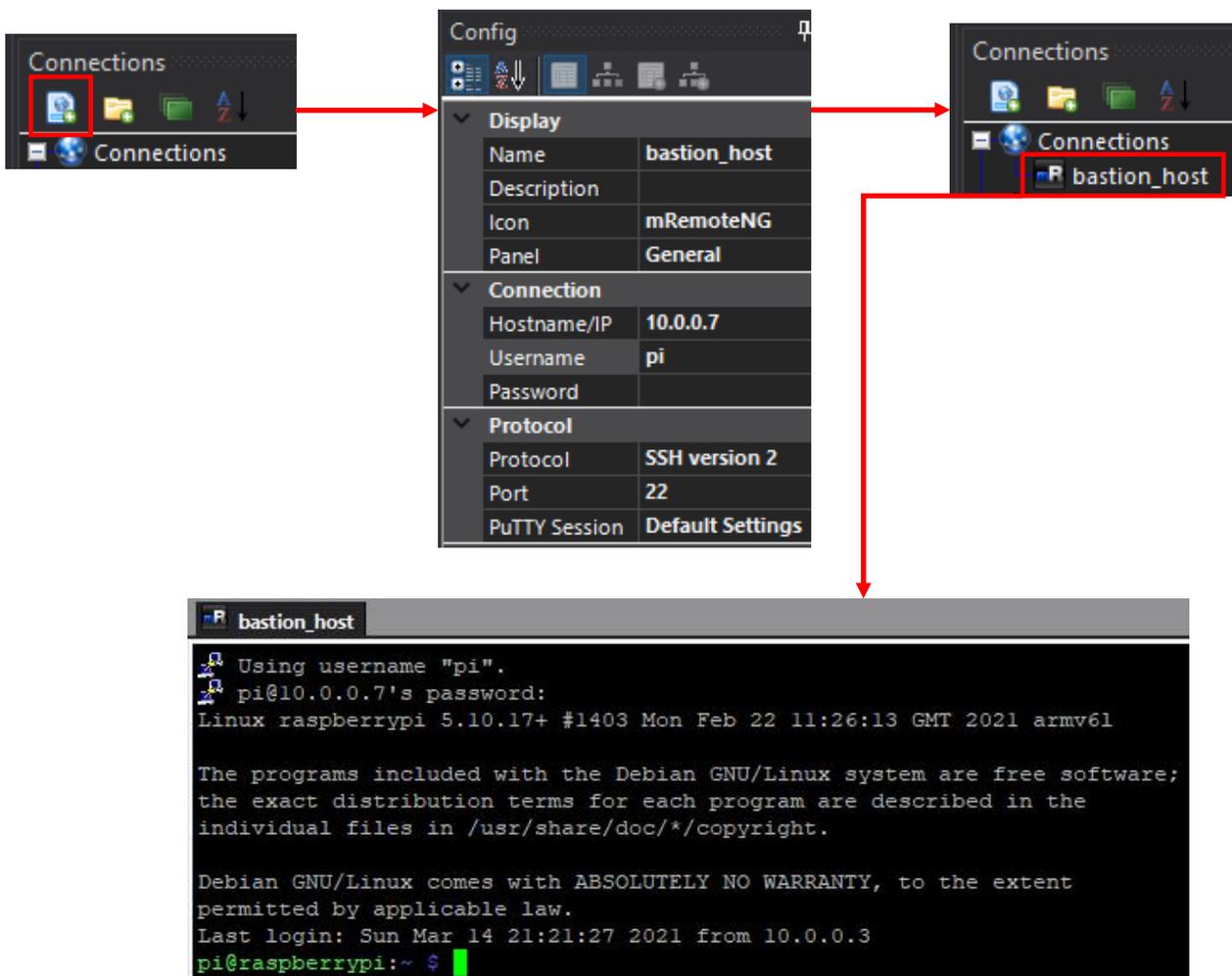
Name	bastion_host
Hostname/IP	10.0.0.7
Username	pi
Protocol	SSH version 2

Note: Students can enter the password for their user into the Password field if desired, but considering we'll be enabling key-based authentication soon (with the option to disable password authentication entirely) I wouldn't bother. If you don't plan on enabling key-based authentication, make sure you enable the configuration options to encrypt and password protect the mRemoteNG connection file. Check out Chapter 15, Section 15.2.3.3 (pp. 775-783). Specifically, the sidebar conversation, *What if I don't want to use 2FA with mRemoteNG?*

If students are not familiar with mRemoteNG, and need some extra guidance, check out chapter 15, sections 15.2.1, and 15.2.2 (pp. 743-749) for further instructions on how to create connection profiles, and how to change the appearance of the UI.

With the first connection profile created, double click on the `bastion_host` profile in the *Connections* pane to test SSH connectivity to the bastion host. If successful, students should be prompted to enter the password of the user account on the bastion host they specified in their connection profile. Enter the password, hit enter, and the connection should complete successfully.

Note: If you're like me and your virtual machine lab environment, and physical network undergoes a lot of changes frequently, you might get a pop-up labeled *puTTYNG Security Alert* on your first connection attempt, complaining about SSH host keys. Don't panic! Check out the sidebar conversation in section 15.2.2, *SSH Host Keys and You* on page 750. To make a long story short, if you see this notification pop up, click *Yes* to continue.



16-30: The first step for Windows users is to open up mRemoteNG, and create a connection profile for the bastion host they created and configured earlier in this chapter. Fill out the Name, Hostname/IP, Username, and Protocol fields, editing the Hostname/IP and Username fields as necessary. When finish, double click on the `bastion_host` connection profile icon under *Connections*, and confirm successful SSH connection.

16.5.2 Enabling SSH Tunneling via PuTTY Session

After confirming initial connectivity to the bastion host, students may type `exit`, or right click on the `bastion_host` tab in the main window, and select *Disconnect*. Next up, students need to enable the SSH forward and dynamic tunnels. To do so, select *Options > Advanced* from the navigation menu, then select *Advanced* from the *mRemotNG Options* menu. Finally, click the *Launch PuTTY* button. The *PuTTYNG Configuration* window will appear.

In the left pane, labeled *Category*, click the `+` symbol next to *SSH*, then select the option labeled *Tunnels*. Let's begin by creating our first forward tunnel. Under the section labeled *Add new forwarded port*, in the *Source port* input box, pick any TCP port number between 8,000 and 65535. As an example, I'm going to choose port `9000`, and enter it into the input box. Next, for the destination, we'll begin by creating a forward tunnel to the SIEM VM. To do so, students will need the IP address of the SIEM virtual machine, and the port number to forward the connection to. In my case, the IP address of the SIEM VM is `172.16.1.3`, and the port number will be `22`, for the SSH service. So, in the *Destination* input box, enter `172.16.1.3:22`, and leave the radio buttons in their default positions – *Local*, and *Auto*. Next, click the *Add* button.

With that out of the way, create two more forward tunnels with the following settings:

Local Port	Destination
9001	172.16.1.4:22
9002	172.16.2.2:22

Note: If necessary, substitute the IP addresses `172.16.1.4` and `172.16.2.2` With the IP addresses of the IPS and Kali virtual machines, if students are using an alternate network configuration for their lab environment.

That leaves us with one more tunnel to create, the dynamic tunnel. To create a dynamic tunnel, enter port number between 8000 and 65535 into the *Source port* input box. As an example, I will use port `9003/TCP`. Do not put anything into the *Destination* input box. Change the first radio button from *Local* to ***Dynamic***, then click the *Add* button. If students performed all of these tasks correctly, the *Forwarded ports* pane will have the following entries:

```
L9000 172.16.1.3:22
L9001 172.16.1.4:22
L9002 172.16.2.2:22
D9003
```

When finished, click on the *Session* option under the *Category* pane. In the *Saved Sessions* input box, name this session `Bastion_Host_Tunnels`, then click the *Save* button. Our new session should appear in the window pane below, under the *Default Settings* session. When this task has been completed, Click *Close* to exit *PuTTYNG Configuration*, followed by *OK* in *mRemoteNG Options*.

Next, students need to apply this PuTTY session to the `bastion_host` mRemoteNG connection profile. In the *Connections* pane, click on the `bastion_host` connection profile to highlight it. Then in the *Config* pane, location the *PuTTY Session* field under the *Protocol* section. Click the downward facing arrow, and select the `Bastion_Host_Tunnels` session. Next, students should double-click on the `bastion_host` connection profile and confirm they are still able to log in to their bastion host VM. With the SSH session connection, open up a command prompt window, and run the command:

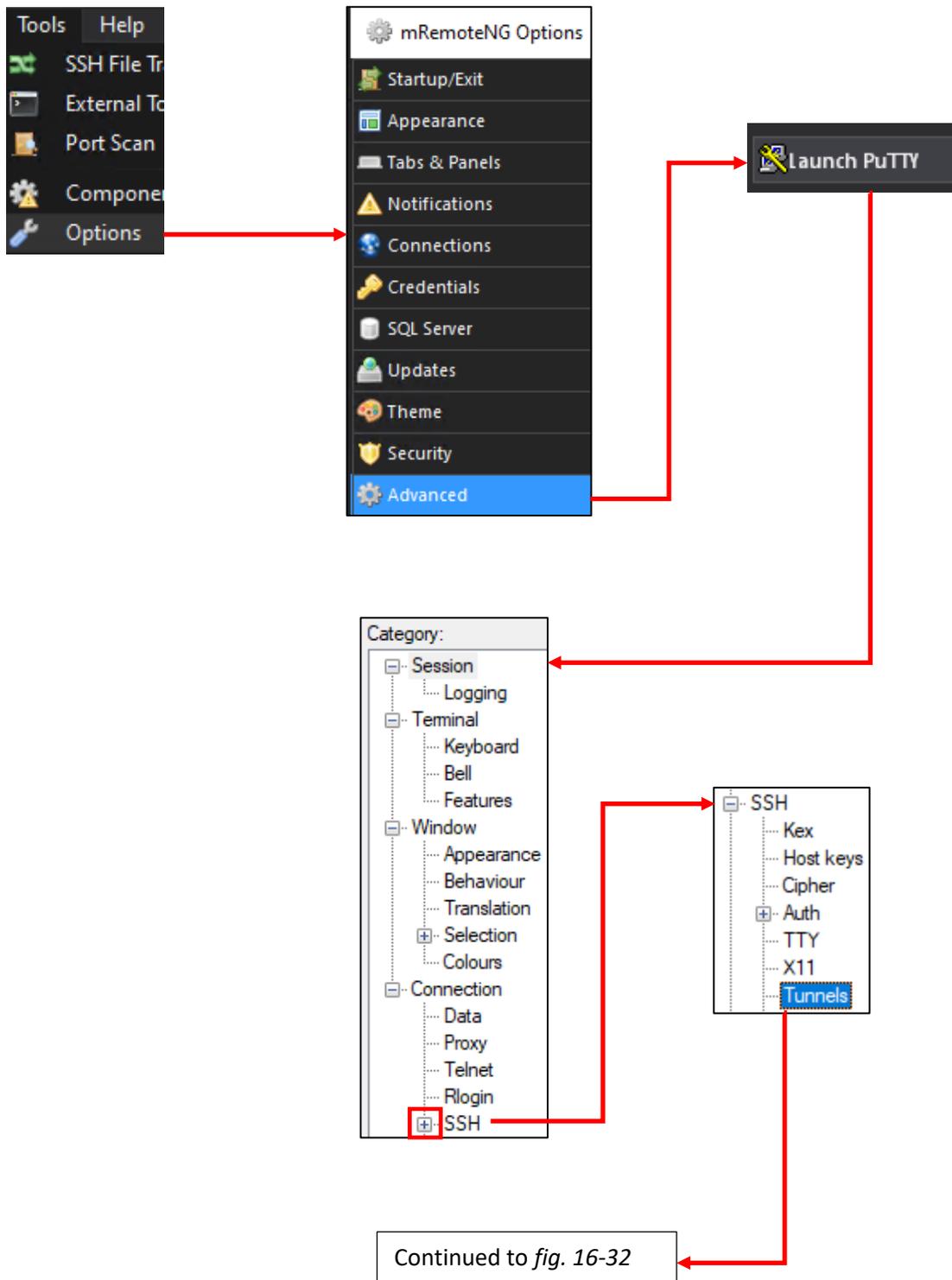
```
netstat -ano | findstr LISTENING | findstr :900
```

This string of commands runs the command `netstat` to show active network connections on Windows. We then use a pipe character ("`|`") to redirect the output of `netstat` to a utility called `findstr`. We tell `findstr` to look for the string "LISTENING". We want to open ports that are listening for a connection on our Windows system. Finally, we run `findstr` again, and tell it to look for the string "`:900`". This will return `9000,9001, 9002`, etc. This is the fastest way to have `findstr` attempt to find all of the tunnels we established on ports `9000` through `9003`. Of course, if students choose to use different source ports to set up their forward and dynamic tunnels, they will have to substitute for those port numbers in the second `findstr` command, instead.

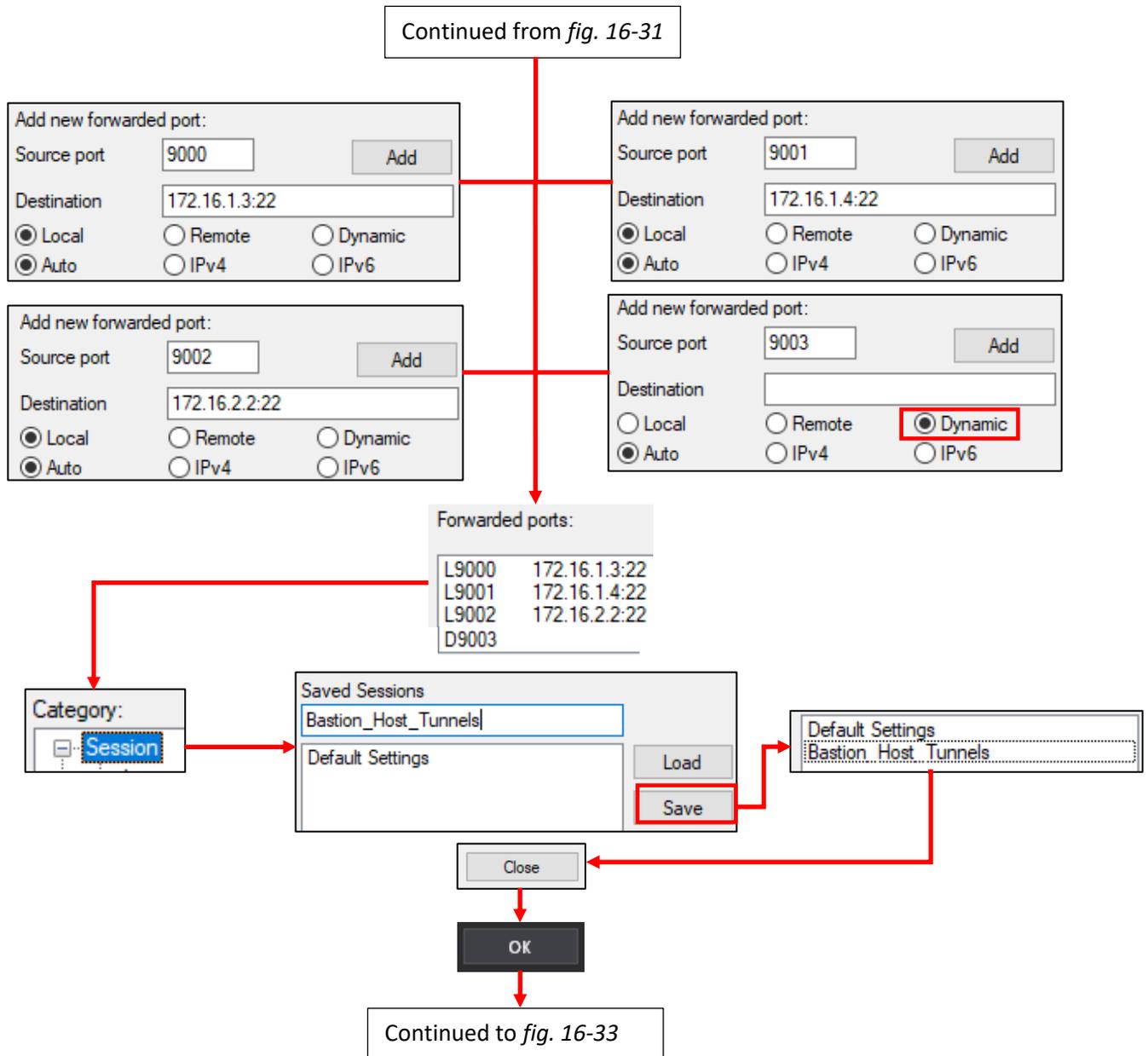
If students did everything correctly and all of the tunnels are up, there will be 8 lines that appear. A listener for ports `9000` through `9003` on `127.0.0.1` (IPv4 loopback address), and A listener for ports `9000` through `9003` on `:::1` (IPv6 loopback address). With the tunnels up and ready to accept connections, our next step will be creating mRemoteNG connection profiles that will utilize the forward tunnels on the bastion host to connect to the SIEM, IPS, and Kali virtual machines. As for the dynamic tunnel, we'll come back to that later in this chapter.

What's a Loopback Address?

A loopback address is a special IP address that refers to the host itself. Usually, this loopback address is assigned to a special loopback virtual interface. This interface is assigned either an IPv4 address on the `127.0.0.0/8` network (usually `127.0.0.1`), or for IPv6, an address in the `:::1/128` range (usually `:::1`). Loopback interfaces and IP addresses are special in that they are not accessible from any external host, just the local computer itself. So, the forward and dynamic tunnel listeners we created are only accessible to applications running on the Windows workstation itself.



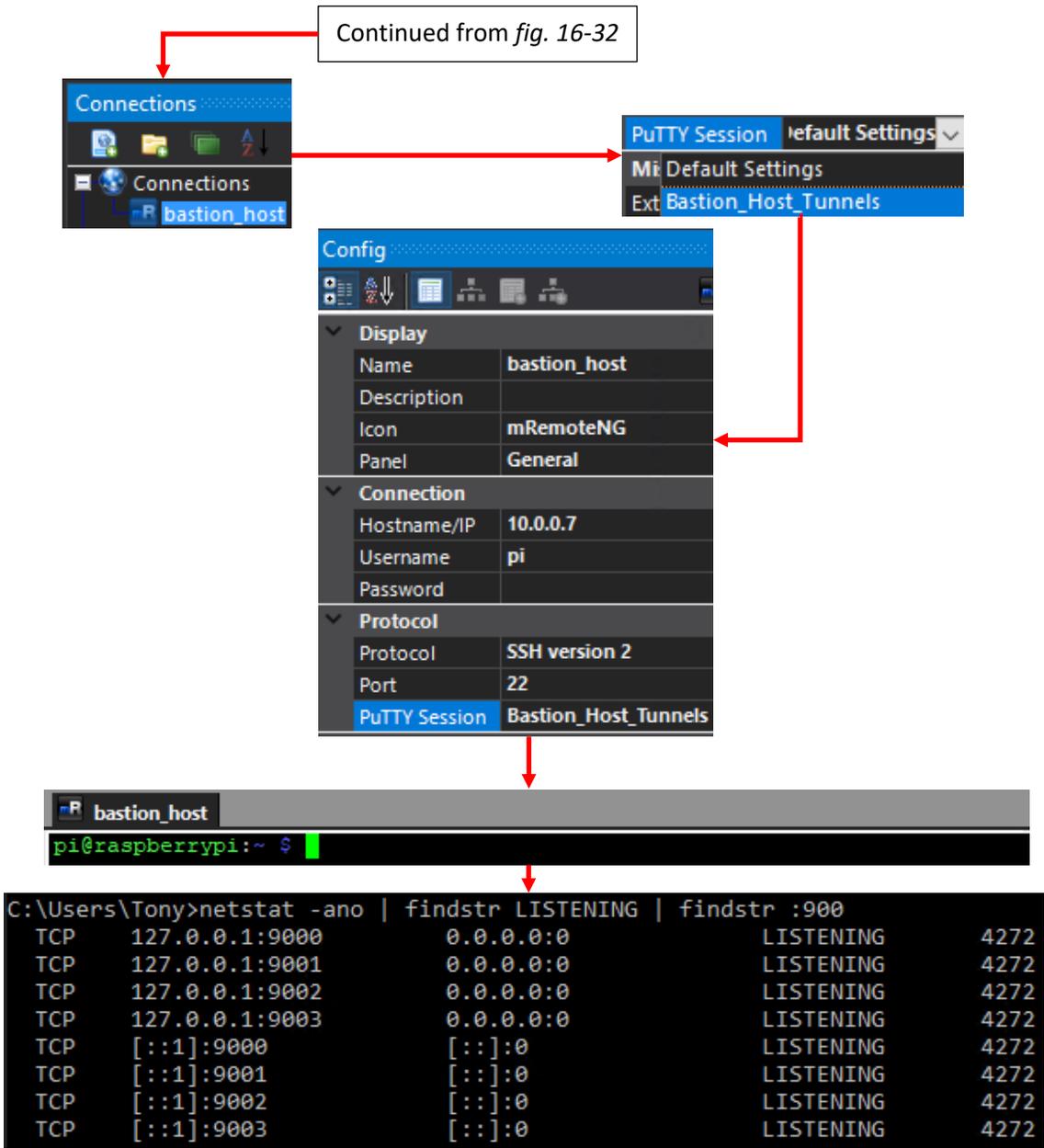
16-31: To enable SSH tunnels on the bastion host, students will need to create a custom PuTTY session. To do so, select *Tools > Options* in the mRemoteNG navigation menu, then in the mRemoteNG Options window, select *Advanced*, and click the *Launch PuTTY* button. In the *Category* pane on the left, scroll down to *SSH*, click the "+" icon, then click the *Tunnels* option.



16-32: In the *Tunnels* configuration option, under *Add new forwarded port* section, students will need to add three forward tunnels – one for the SIEM VM (172.16.1.3) on port 22/TCP, one for the IPS VM (172.16.1.4) on port 22/TCP, and another for the Kali VM (172.16.2.2) on port 22/TCP as well. Additionally, students will need to create one Dynamic tunnel for this session as well. When finished, There should be four entries in the *Forward ports* pane:

```
L9000 172.16.1.3:22
L9001 172.16.1.4:22
L9002 172.16.2.2:22
D9003
```

When finished, Navigate to *Session*, and save this PuTTY session as *Bastion_Host_Tunnels*, then exit both the *PuTTYNG Configuration*, and *mRemoteNG Options* windows by clicking the *Close* followed by the *OK* button.



16-33: With the new PuTTY session created, students need to apply it to the bastion host's mRemoteNG connection profile. To do so, highlight the bastion_host connection profile under the *Connections* pane, then under the *Config* pane in the *Protocol* section, modify the *PuTTY Session* from *Default Settings* to the *Bastion_Host_Tunnels* profile. Next, double click on the bastion_host connection profile to create an SSH connection to the bastion host. With the SSH session established, open the Windows command prompt and run:

```
netstat -ano | findstr Listening | findstr :900
```

This series of commands looks for Listening network sockets on the local Windows system, and isolates TCP listening sockets start with ":900". If the forward and dynamic tunnels are all working properly and waiting for connections, there should be eight entries in total. Four listeners on ports 9000-9003 on 127.0.0.1 (IPv4 loopback address), and four more listeners on ports 9000-9003 on ::1 (IPv6 loopback address)

16.5.3 Connecting to the SIEM, IPS and Kali VMs using Forward Tunnels

Now that students are connected to their bastion host, and have SSH tunnels waiting for connections, the next step is creating mRemoteNG connection profiles to connection to the SIEM, IPS, and Kali virtual machines, through the forward tunnels we just created. Create three mRemoteNG connection profiles with the following settings:

Name	SIEM	IPS	Kali
Hostname/IP	127.0.0.1	127.0.0.1	127.0.0.1
Username	Username created during OS installation	Username created during OS installation	Username created during OS installation
Password	Blank	Blank	Blank
Protocol	SSH version 2	SSH version 2	
Port	9000	9001	9002

Students may leave the other fields at their default values for now. After all three connection profiles have been created, double-click on each of them to confirm SSH connectivity to the SIEM, IPS and Kali virtual machines, respectively. Just to reiterate, please be aware that **students must have an active SSH connection to their bastion host with the Bastion Host Tunnels PuTTY session defined. in order to test SSH connectivity to the SIEM, IPS and Kali virtual machines.**

Forwarded ports:

L9000	172.16.1.3:22
L9001	172.16.1.4:22
L9002	172.16.2.2:22

SIEM Profile Configuration:

- Name: SIEM
- Description:
- Icon: mRemoteNG
- Panel: General
- Connection Hostname/IP: 127.0.0.1
- Username: ayy
- Password:
- Protocol Port: 9000
- PuTTY Session: Default Settings

SIEM Terminal Output:

```

ayy@siem:~$ whoami; hostname; ip -br addr
ayy
siem
lo                UNKNOWN      127.0.0.1/8 :
ens160            UP           172.16.1.3/24
  
```

IPS Profile Configuration:

- Name: IPS
- Description:
- Icon: mRemoteNG
- Panel: General
- Connection Hostname/IP: 127.0.0.1
- Username: ayy
- Password:
- Protocol Port: 9001
- PuTTY Session: Default Settings

IPS Terminal Output:

```

ayy@ips:~$ whoami; hostname; ip -br addr
ayy
ips
lo                UNKNOWN      127.0.0.1/8 :
ens160            UP           172.16.1.4/24
ens192            DOWN
ens224            DOWN
  
```

Kali Profile Configuration:

- Name: Kali
- Description:
- Icon: mRemoteNG
- Panel: General
- Connection Hostname/IP: 127.0.0.1
- Username: ayy
- Password:
- Protocol Port: 9002
- PuTTY Session: Default Settings

Kali Terminal Output:

```

ayy@Kali:~$ whoami; hostname; ip -br addr
ayy
Kali
lo                UNKNOWN      127.0.0.1/8 :
eth0              UP           172.16.2.2/24
  
```

16-34: Next, students will create three mRemoteNG connection profiles for the SIEM, IPS, and Kali virtual machines. Pay special attention to the *Port* field, under the *Protocol* section. **The port number students enter MUST match the ports for the forward tunnels configured for SIEM, IPS, and Kali VMs in the bastion_host tunnel PuTTY session.** After creating the connection profiles, students should connect to their bastion_host system, then verify successful connectivity to the SIEM, IPS, and Kali virtual machines.

16.5.4 Generating SSH Keys for Key-Based Authentication (Optional)

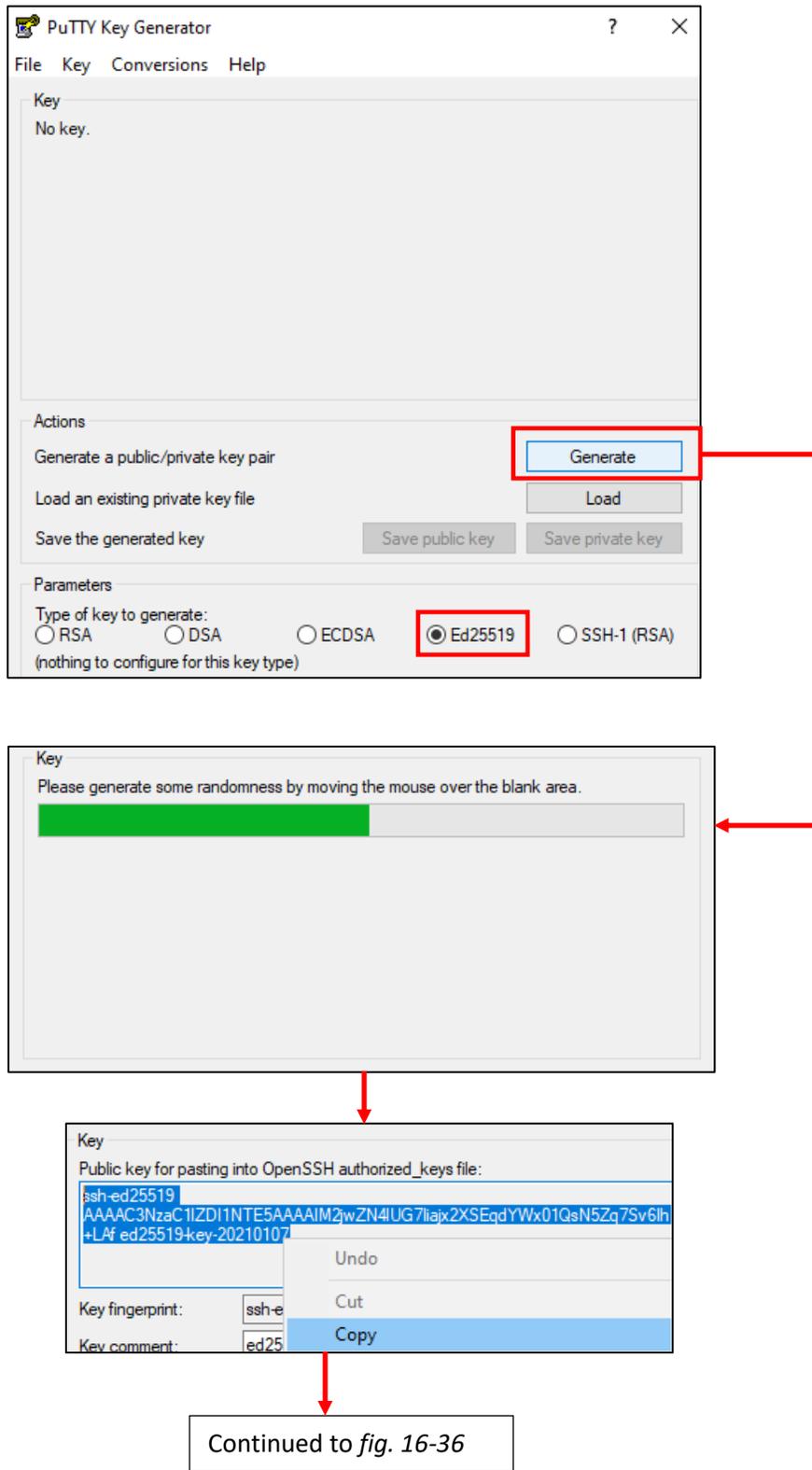
In this section, students will use either PuTTYgen (or the `ssh-keygen` command) to generate public/private cryptographic key pair in order to enable key-based authentication to their lab virtual machines, and bastion host. As the title of the section implies, configuring key-based authentication for SSH is entirely optional, but highly recommended.

This topic has been covered extensively Chapter 15, [section 15.2.3](#) (pp. 750-759), and should be completely identical for students looking to generate SSH keys for use with a bare-metal lab environment. Below is a summarization of the steps:

- Download and run `puttygen.exe`
- Generate an ed25519 key pair
 - **Optional:** assign a passphrase to the private key if interested in establishing two-factor authentication to lab virtual machines and bastion host. Students should save the ssh private key passphrase to a password manager.
- In the *PuTTYgen* window, there is a box labeled *Public key for pasting into OpenSSH authorized_keys file*, beginning with "ssh-ed25519". Copy the contents of this box to *Notepad++* (or any other Windows text editor that allows students to convert the end of line characters to UNIX/Linux format). **Ensure that the number of lines in the notepad++ is 1. In total.** (Notepad++ displays the line numbers of a text file in the left margin). In the navigation menu, select *Edit > EOL Conversion > Unix (LF)*.
- Select *File > Save As*, and save the file as `authorized_keys` with **no** file extension (select the *All Files (*.*)* file extension). Save this file in the same location as the private key file for easy access.

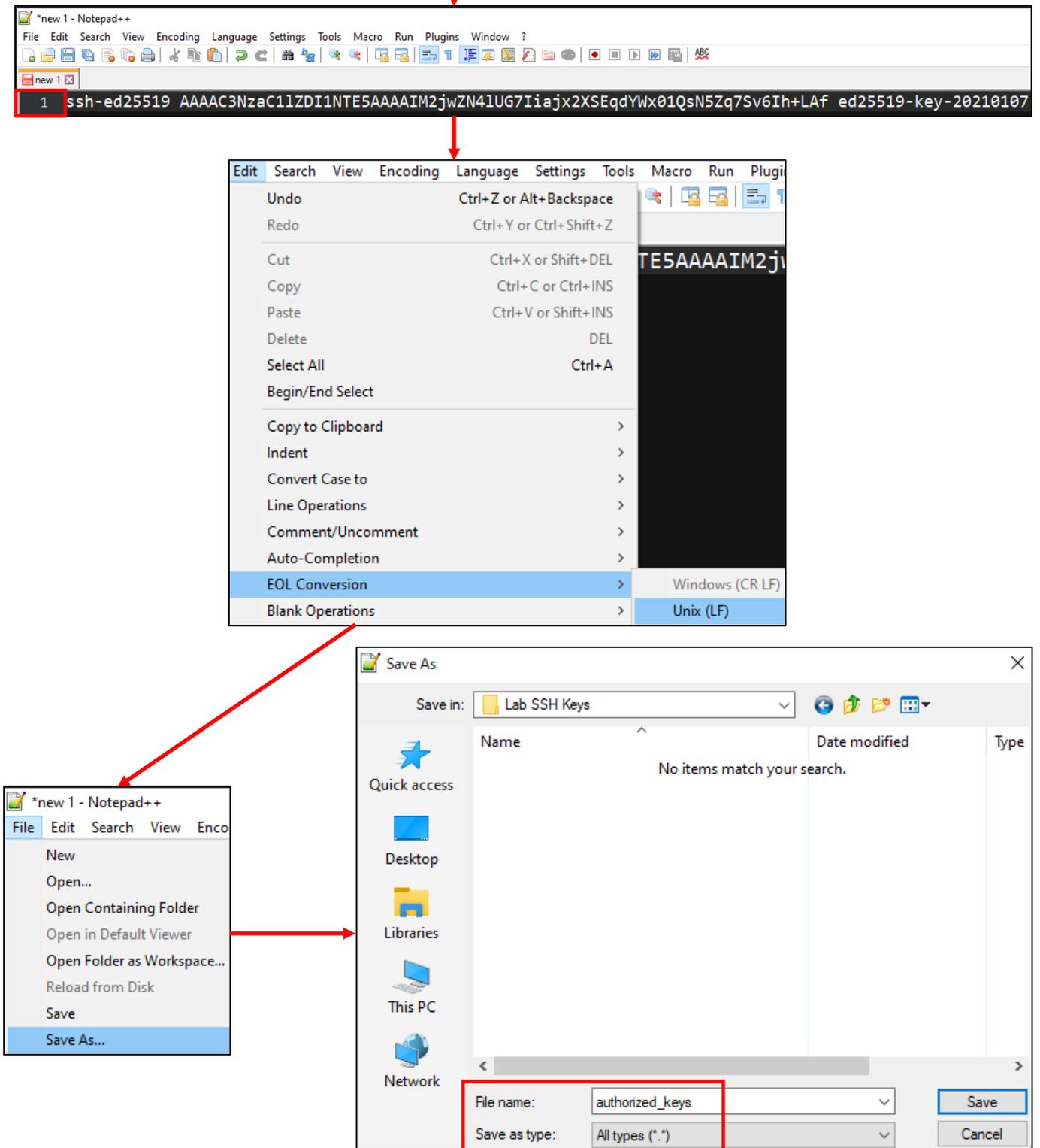
For the visual learners, I copied illustrations 15-15 through 15-18 from Chapter 15 down below for your convenience.

Note: The sidebar discussion in section 15.2.3, [Alternate Key Generation Method](#) (pp. 756-759) may also be used for generating/converting the SSH keys for your bare-metal lab, if desired. Also, as a side note, if you've generated an SSH public/private key pair for a hosted hypervisor lab environment, and want to re-use that key pair, that is perfectly fine.

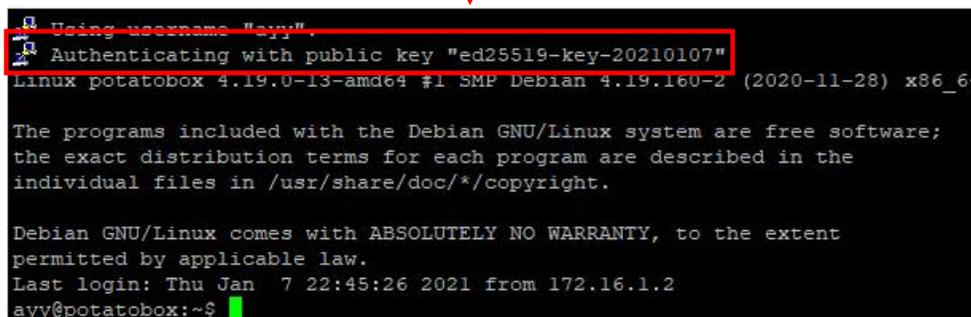
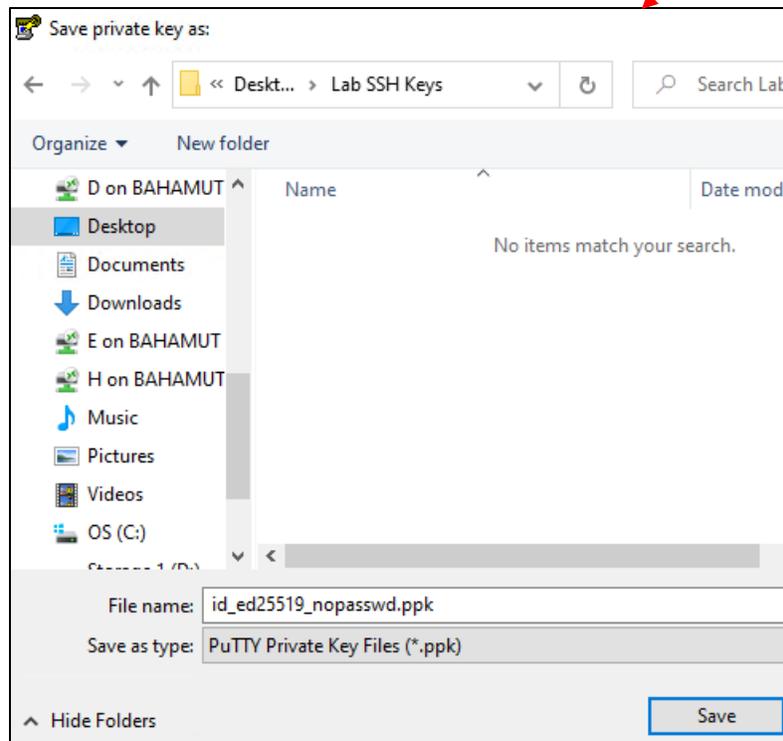
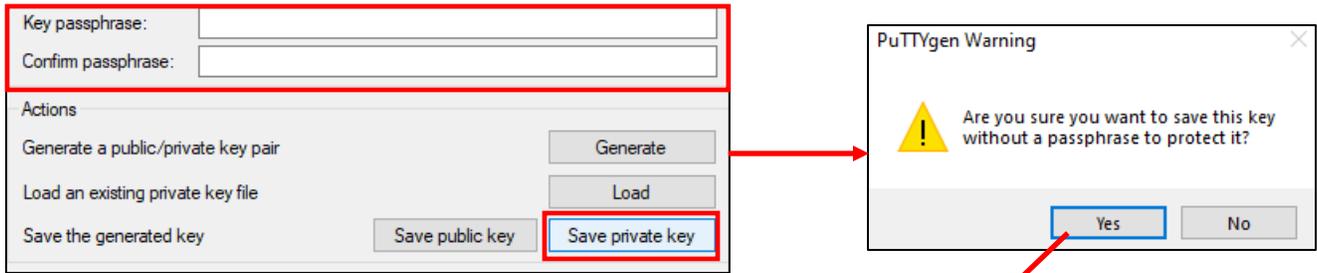


16-35: Open puttygen.exe, Click on the *Ed25519* radio button under *Parameters*, then click on the *Generate* button under *Actions*. Move your mouse cursor under the *Key* section of the window to create entropy and help generate your SSH key. When finished, copy the entire section under the text *Public key for pasting into OpenSSH authorized_keys file*

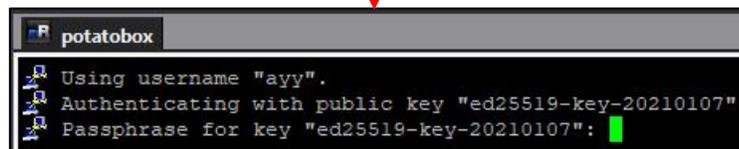
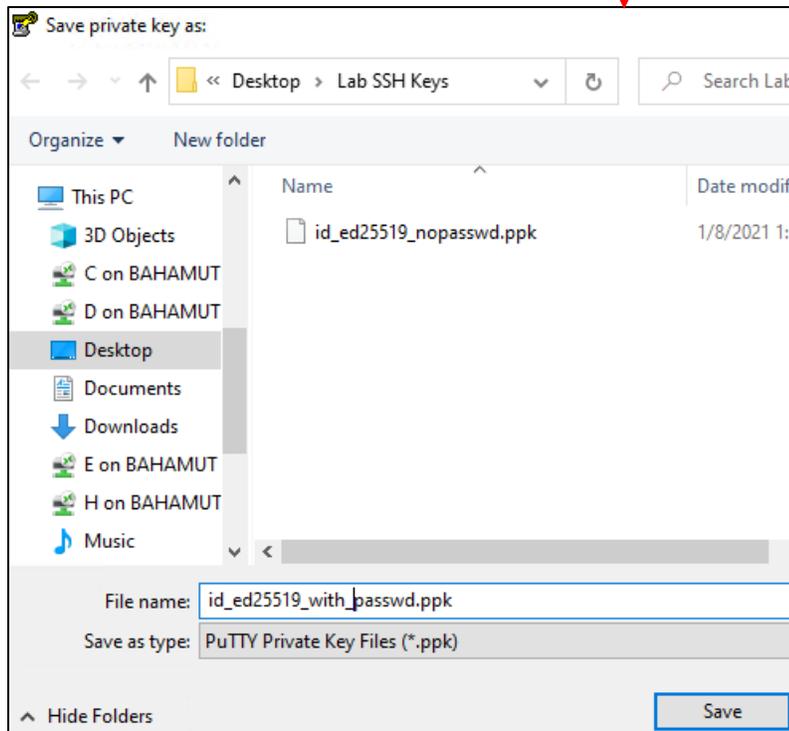
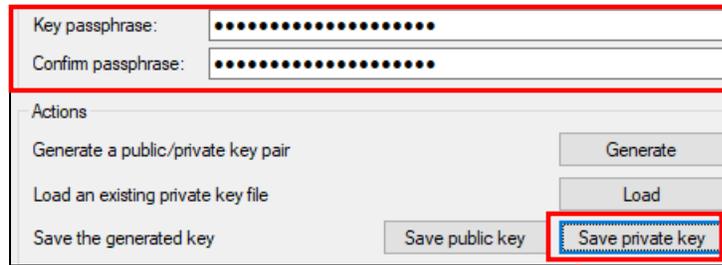
Continued from *fig. 16-35*



16-36: Next, paste the SSH public key copied from puttygen into Notepad++. Using the line numbers on the left margin of the Notepad++ window, ***make sure the file is only a single line long***. Afterwards, Select *Edit > EOL Conversion > Unix (LF)*. Finally, select *File > Save As*, and in the *File name* input box, enter `authorized_keys`, and in the *Save as type* drop-down menu, select *All types (*.*)*. Be sure to save this file to an easy to access location.



16-37: If students click the *Save private key* button under *Actions* with the *Key* and *Confirm passphrase* input boxes blank, this allows the creation of an SSH private key that is not password protected. puttygen will ask the user to confirm if this is what they want to do. If students click *Yes*, they are prompted to save their private key. The final image on this page is of an SSH session configured to use key-based authentication using a private key that is not password protected. mRemoteNG notifies the user that the the public/private key pair is being used to authenticate, and once successful, a session is instantly granted.



16-38: In comparison, if students elect to set a password on the private key, then click *Save private key*, and configure mRemoteNG to use the password-protected key, students will be prompted to enter that key's password when they log in. It's a very good idea to store the password to the private key in a password manager (e.g., KeePassXC)

16.5.5 Copying The authorized_keys File to the Bastion Host, and Lab VMs

With a properly formatted SSH public and private key pair, the next step involves transferring the public key file, `authorized_keys`, to the SIEM, IPS, and Kali VMs, as well as the bastion host. In chapter 15, [section 15.2.3.2 \(pp. 760-774\)](#), I described three methods for transferring the contents of the `authorized_keys` file to a remote host:

1. Using WinSCP and the SCP protocol to copy the existing `authorized_keys` file from students' Windows workstation to the remote system.
2. Copying the contents of the `authorized_keys` file to the Windows clipboard, then using an SSH session to the remote system, the `echo` command, output redirection, and pasting the contents of the file to the command line to write the `authorized_keys` file on the remote host.
3. Copying the contents of the `authorized_keys` file to the Windows clipboard, then using an SSH session to the remote system, the `vi` text editor, and pasting the contents of the file to the editor to write the `authorized_keys` file on the remote host.

The only method that differs in any major way from the methods described in Chapter 15 is method number 1 – WinSCP. As with [section 16.5.4](#), I will provide a brief outline of the tasks students must perform to successfully copy their public key their lab systems, including the bastion host.

Note: Students will need SSH connectivity to the bastion host, SIEM, IPS, and kali VMs to perform any of the methods listed above. For the SIEM, IPS, and kali virtual machines, that means ensuring that the SSH forward tunnels (provided from the *Bastion_Host_Tunnels* PuTTY session we configured earlier) are operating properly.

16.5.5.1: Method 1 – WinSCP

- Begin by opening an SSH session to the bastion host system, then open an SSH sessions to the SIEM, IPS, and kali virtual machines, through the bastion host's forward tunnels
- Open the WinSCP application, and if the *Login* window is not displayed, click the *New Session* icon
- In the *Login* window, students should click *New Site*, and in the *Session* section, enter the Host name (IP address), Port number, and User name they configured for their bastion host system. As an example, this is what I entered for my raspberry pi:

System	bastion host
Host name	10.0.0.7
Port	22
User name	pi
Password	[leave blank]

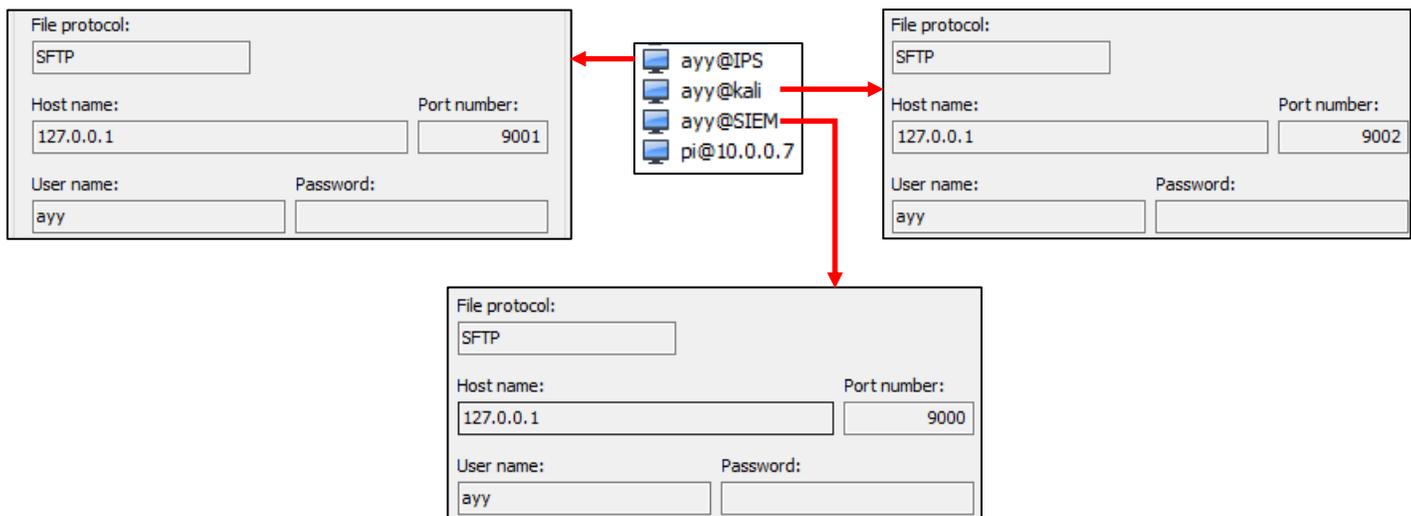
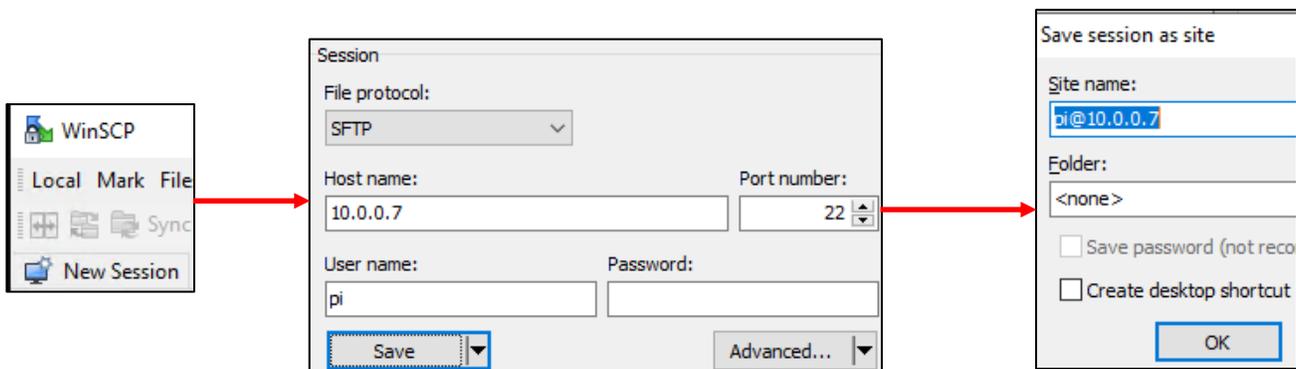
- Click the *Save* button, and in the pop-up box that appears, name the WinSCP connection in the *Site name* field. (e.g., WinSCP named my session as pi@10.0.0.7), then click *OK*
- Create additional WinSCP site profiles for the SIEM, IPS, and Kali virtual machines, using the table below as a guide:

Virtual Machine	SIEM	IPS	Kali
Host name	127.0.0.1	127.0.0.1	127.0.0.1
Port	9000	9001	9002
User name	[SIEM username]	[IPS username]	[Kali username]
Password	[leave blank]	[leave blank]	[leave blank]
Site Name	[username]@SIEM	[username]@IPS	[username]@kali

- Open a WinSCP session to bastion host by double-clicking its site name (e.g., pi@10.0.0.7). Upon connecting, enter the SSH password for the bastion host user to connect.

Note: students might be prompted about SSH host keys upon attempting to connect to the system for the first time, long story short, click *Yes* to continue. Check out the sidebar conversation [*SSH Host Keys and You \(Part 2\)*](#) p. 764 for more details.

- The left pane represents the local Windows system. Navigate to the directory on the Windows workstation that contains the properly formatted `authorized_keys` file, and copy it over to the remote system's home directory (e.g., `/home/[username]` -- `/home/pi` in my case), by dragging the `authorized_keys` file from the left pane to the right pane. Close the SSH session by clicking the "X" next to the bastion host site name.
- Repeat this process for the SIEM, IPS, and kali virtual machines.
- With the `authorized_keys` file successfully copied to each system, run the following commands on the bastion host, SIEM, IPS and kali SSH sessions:
 - `mkdir ~/.ssh`
 - `chmod 700 ~/.ssh`
 - `mv ~/authorized_keys ~/.ssh`
 - `chmod 600 ~/.ssh/authorized_keys`

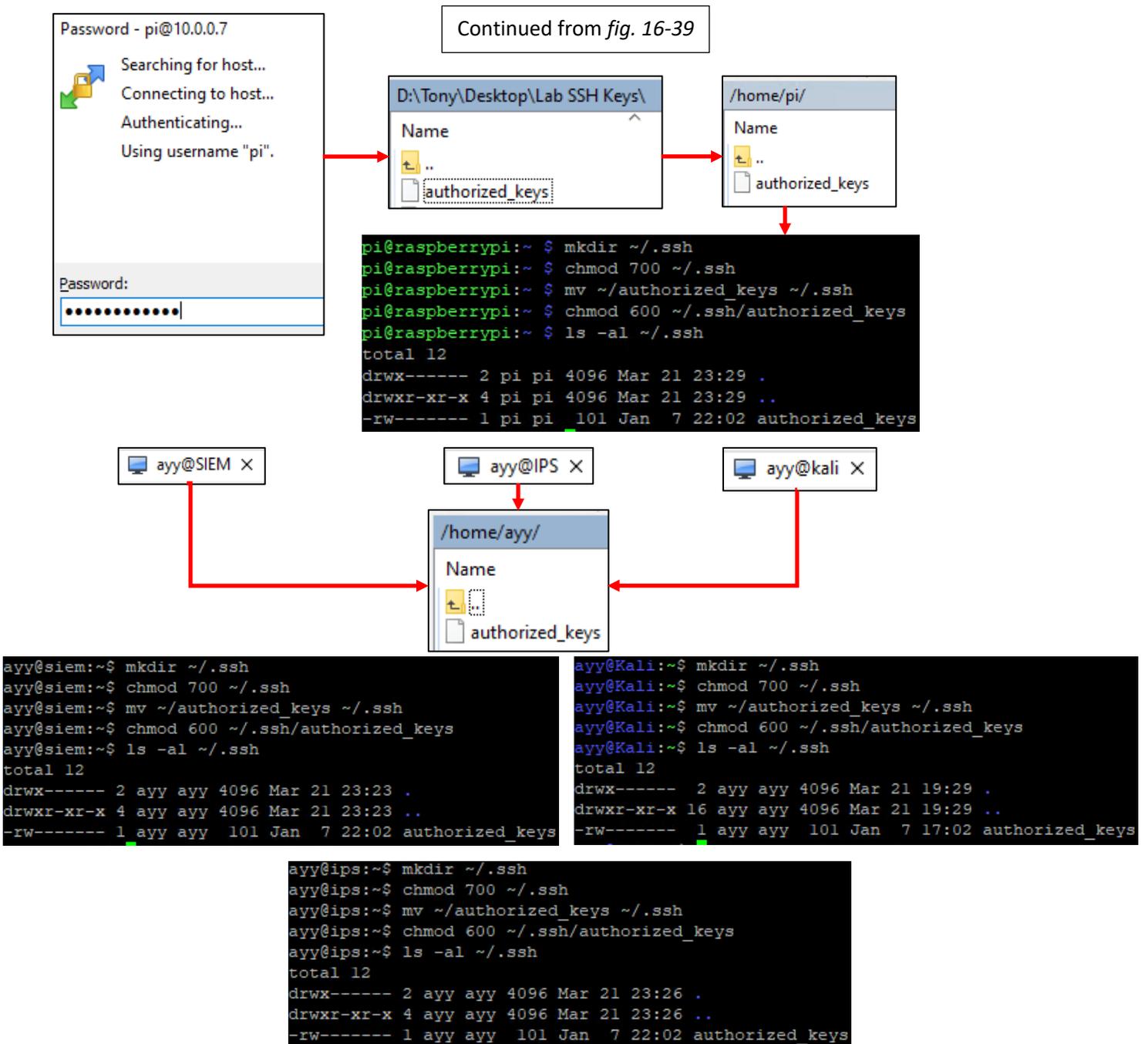


Continued to *fig. 16-40*

16-39: To transfer the authorized keys file via SCP, begin by opening an SSH session to the bastion host system, SIEM, IPS and Kali virtual machines. Run the commands:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
```

Open WinSCP and in the *Login* window (can be accessed by clicking *New Site*), enter connection information for the bastion host system. Click *Save*, and assign a *Site name* to the bastion host session. Repeat this process, for the SIEM, IPS, and Kali VMs. However, instead of using the actual IP address, the IP address for all three entries should be 127.0.0.1, and the port numbers should be the exact same port numbers used to connect to the virtual machines over SSH via forward tunnels. Refer to *fig. 16-34*.



16-40: Open a WinSCP to the bastion host. Locate the formatted `authorized_keys` file on the Windows workstation in the left pane, then drag it over to the right to transfer it over the bastion host user's home directory. In an SSH session to the bastion host in mRemoteNG, run the following commands:

```

mkdir ~/.ssh
chmod 700 ~/.ssh
mv ~/authorized_keys ~/.ssh
chmod 600 ~/.ssh/authorized_keys
ls -al ~/.ssh
  
```

Repeat this process for the SIEM, IPS and Kali virtual machines.

16.5.5.2: Method 2 – Copy, Paste, echo, and file redirection

- Begin by opening an SSH session to the bastion host, SIEM, IPS, and kali VMs
- Locate the properly formatted `authorized_keys` file, and open it in `Notepad++`
- Highlight all of the text in the document (This can be done via the `Edit` option in the context menu, and choose the `Select All` option. There is also a `Select All` option in the right-click context menu, or the `Ctrl+a` keyboard shortcut)
- With all of the text highlighted, `Copy` the text from the document (Again, this can be done via `Edit > Copy`, `right-click > Copy`, or the `Ctrl+c` keyboard shortcut)
- Run the following commands on the bastion host, SIEM, IPS and kali SSH sessions:
 - `mkdir ~/.ssh`
 - `chmod 700 ~/.ssh`
 - `echo "[right click on the putty session one time]" >> ~/.ssh/authorized_keys`
 - `chmod 600 ~/.ssh/authorized_keys`

The diagram shows the following steps:

- Opening `authorized_keys` in `Notepad++`.
- Selecting all text in the file.
- Copying the text.
- Running the following commands in terminal sessions on the bastion host, SIEM, IPS, and Kali virtual machines:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
echo "[right click on the putty session one time]" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```
- Running `ls -al ~/.ssh` on each machine to verify the file's presence and permissions.

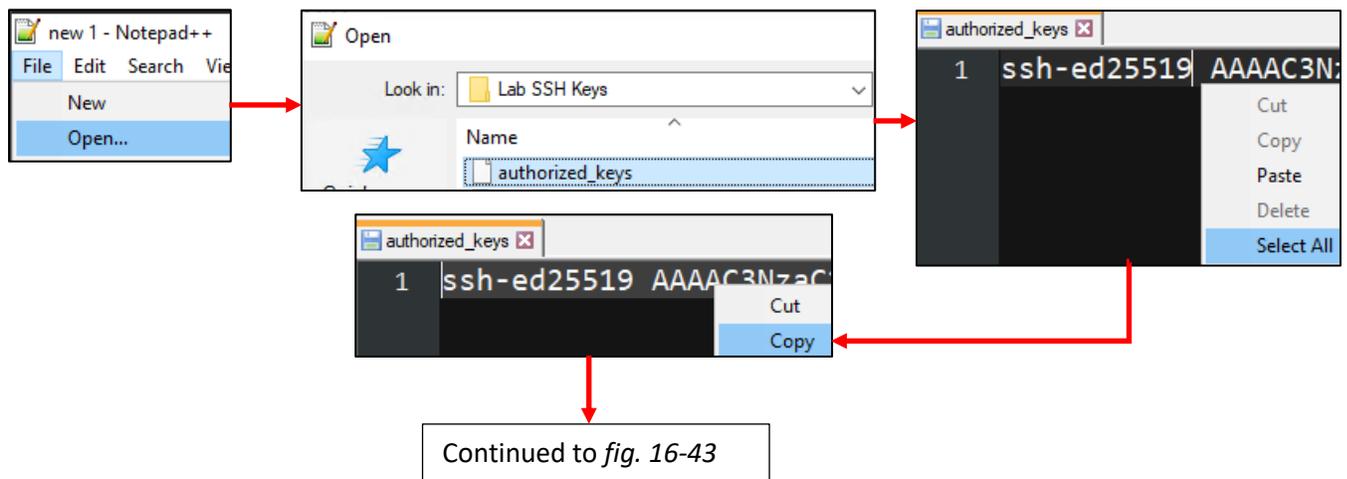
16-41: Open the `authorized_keys` file in `Notepad++`. `Select all`, then `Copy` all the file contents. Next, run the following commands in SSH sessions on the bastion host, SIEM, IPS, and Kali virtual machines:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
echo "[right click on the putty session one time]" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Right-clicking in the terminal session should paste the file contents to the SSH session. **Make sure the contents of the authorized keys file is pasted between the quotation marks in the echo command.**

16.5.5.3: Method 3 – Copy and Paste, using vi

- Begin by opening an SSH session to the bastion host, SIEM, IPS, and kali systems
 - Locate the properly formatted `authorized_keys` file, and open it in *Notepad++*
 - Highlight all of the text in the document (This can be done via the *Edit* option in the context menu, and choose the *Select All* option. There is also a *Select All* option in the right-click context menu, or the *Ctrl+a* keyboard shortcut)
 - With all of the text highlighted, *Copy* the text from the document (Again, this can be done via *Edit > Copy*, *right-click > Copy*, or the *Ctrl+c* keyboard shortcut)
 - Run the following commands on the bastion host, SIEM, IPS, and kali SSH sessions:
 - `mkdir ~/.ssh`
 - `chmod 700 ~/.ssh`
 - `vi ~/.ssh/authorized_keys`
 - In the *vi* session, enter insert mode by hitting the "i" key, then right click to paste the contents of the `authorized_keys` file
- Note:** *vi* on Kali Linux is slightly different. Students will instead have to paste the `authorized_keys` file with the keyboard combination, `shift + ins`.
- After pasting the contents of the file into *vi*, hit the escape key to exit insert mode, then type `:wq!` write the changes to the file, and exit *vi* immediately
 - Finally, run the command:
 - `chmod 600 ~/.ssh/authorized_keys`



16-42: The first steps of this method are the same as method 2: Establish SSH sessions to the bastion host, SIEM, IPS, and kali virtual machines. Open the `authorized_keys` file in *Notepad++*, and copy the contents to the windows clipboard.

Continued from *fig. 16-42*

```
mkdir ~/.ssh  
chmod 700 ~/.ssh  
vi ~/.ssh/authorized_keys
```

```
"~/.ssh/authorized_keys" [New File]
```

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJCNsspsi/fo6o8w7bRf0OR2P7kIXzcqsgOqoJKAJc79 tony@StarFall  
~  
~  
-- INSERT --
```

```
:wq!
```

```
chmod 600 ~/.ssh/authorized_keys
```

```
pi@raspberrypi:~$ ls -al ~/.ssh  
total 12  
drwx----- 2 pi pi 4096 Mar 21 23:29 .  
drwxr-xr-x 4 pi pi 4096 Mar 21 23:29 ..  
-rw----- 1 pi pi 101 Jan 7 22:02 authorized_keys
```

```
ayy@siem:~$ ls -al ~/.ssh  
total 12  
drwx----- 2 ayy ayy 4096 Mar 21 23:23 .  
drwxr-xr-x 4 ayy ayy 4096 Mar 21 23:23 ..  
-rw----- 1 ayy ayy 101 Jan 7 22:02 authorized_keys
```

```
ayy@ips:~$ ls -al ~/.ssh  
total 12  
drwx----- 2 ayy ayy 4096 Mar 21 23:26 .  
drwxr-xr-x 4 ayy ayy 4096 Mar 21 23:26 ..  
-rw----- 1 ayy ayy 101 Jan 7 22:02 authorized_keys
```

```
ayy@kali:~$ ls -al ~/.ssh  
total 12  
drwx----- 2 ayy ayy 4096 Mar 21 19:29 .  
drwxr-xr-x 16 ayy ayy 4096 Mar 21 19:29 ..  
-rw----- 1 ayy ayy 101 Jan 7 17:02 authorized_keys
```

16-43: with the `authorized_keys` file copied, run:

```
mkdir ~/.ssh  
chmod 700 ~/.ssh  
vi ~/.ssh/authorized_keys
```

Enter vi's insert mode by hitting the 'i' key. For the bastion host, SIEM, and IPS SSH sessions, right-clicking in insert mode should paste the contents of the `authorized_keys` file directly into the vi text editor. On the kali SSH session, hit `shit+ins` instead. With the contents copied into the vi session, hit `esc` to exit insert mode, then type `":wq!"` to write the file to disk, and quit vi immediately. Once finished run the command:

```
chmod 600 ~/.ssh
```

Perform this process on the bastion host, SIEM, IPS, and kali SSH sessions.

16.5.6: Creating and Modifying PuTTY Sessions to Enable Key-Based Authentication

So far, students have generated an SSH public/private key pair, and using one of three methods, copied the pre-formatted public key file, `authorized_keys`, to the bastion host, SIEM, IPS, and kali systems. The next step involves creating a custom PuTTY session for the SIEM, IPS, and Kali virtual machines, and modifying the existing *Bastion_Host_Tunnels* session for the bastion host to enable key-based authentication.

Just like the prior sections, this was also covered extensively in Chapter 15. This time in [section 15.2.3.3](#) (pp. 775-783). Let's start by creating a new PuTTY session for the SIEM, IPS and kali virtual machines, by summarizing the instructions from chapter 15:

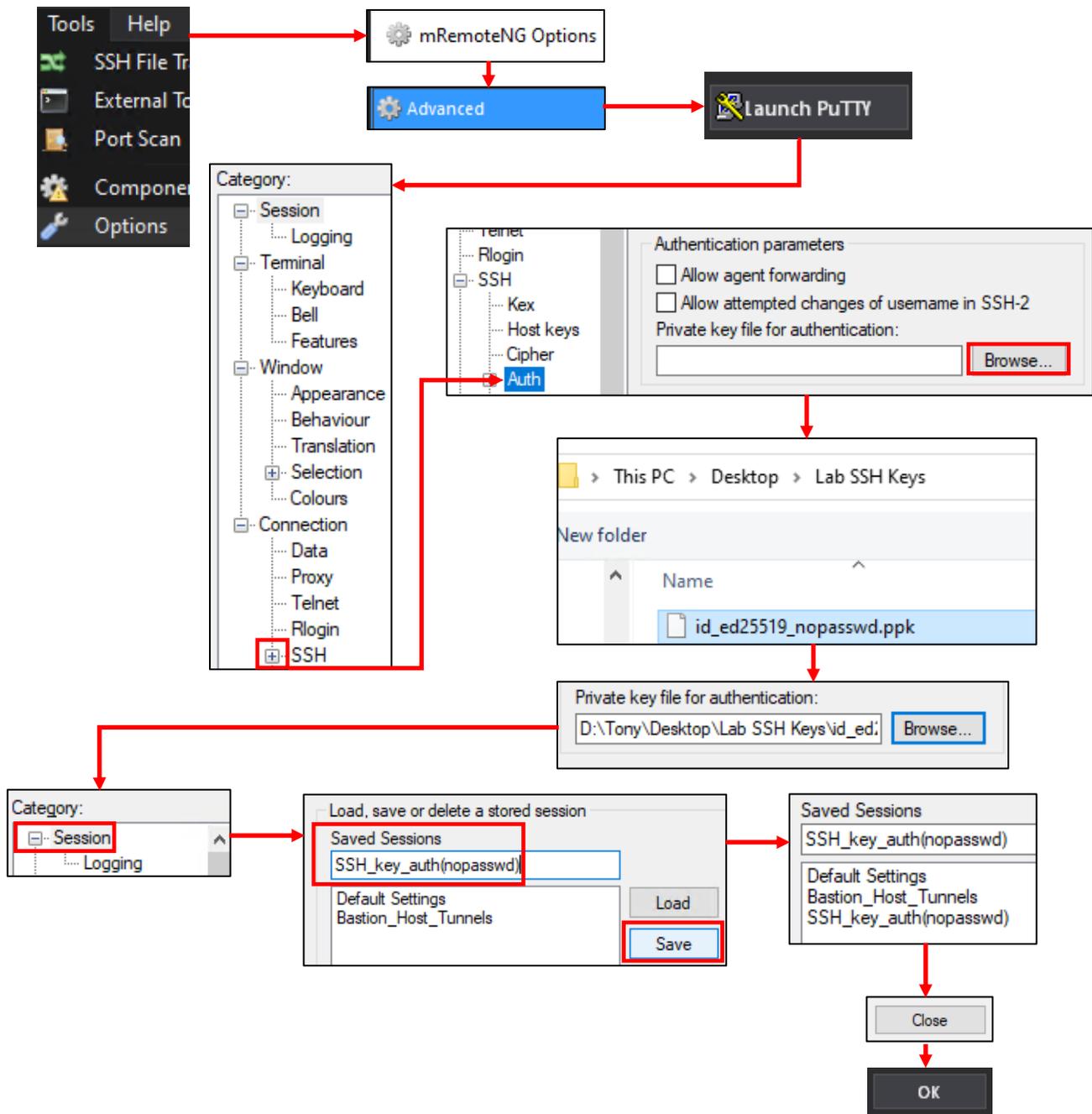
- Open mRemoteNG and access *Tools > Options*. In the *mRemoteNG Options* window, Click *Advanced*, then click the *Launch PuTTY* button
- In the *PuTTYNG Configuration* window, under the *Category* pane, click the "+" symbol next to *SSH*, then click the *Auth* in the sub-menu that appears
- In the *Authentication parameters* section, locate the input box labeled *Private key file for authentication*, and click the *Browse* button to the right of the input box
- An explorer window will appear. Students should browse to the location of the `.ppk` file they created earlier when generating their SSH public/private keypair, and select the `.ppk` file they would like to use
- Back in the *PuTTYNG Configuration* window, scroll back to the top of the *Category* pane, and select *Session*
- In the *Load, save, or delete a stored session* section, input a descriptive name in to the *Saved Sessions* input box (e.g., `SSH_key_auth(nopasswd)` for passwordless key-based auth, or maybe `SSH_key_auth(passwd)` for key-based auth with a password protected SSH private key), then click *Save*. The name of the new session should appear in the saved sessions list along with the Default Settings, and *Bastion_Host_Tunnels* sessions.
- Click *Close* to exit the *PuTTYNG Configuration* window, then *OK* to exit the *mRemoteNG Options* window

Next, to reconfigure the *Bastion_Host_Tunnels* PuTTY session for key-based authentication, follow these instructions:

- Access the *PuTTYNG Configuration* window again, and under the *Category* pane, select *Session*. Under the *Load, save or delete a stored session* section click on the *Bastion_Host_Tunnels* session to highlight it, then click the *Load* button
- Under the *Category* pane, click the "+" symbol next to *SSH*, then click the *Auth* in the sub-menu that appears
- Once again, locate the input box labeled *Private key file for authentication*, and click the *Browse* button to the right of the input box
- Browse to the location of the .ppk file created earlier when generating the SSH public/private keypair, and select it
- Back in the *PuTTYNG Configuration* window, scroll back to the top of the *Category* pane, and select *Session*
- The *Saved Sessions* input box should already display the name *Bastion_Host_Tunnels* in the input box. Click the *Save* button to save changes to the PuTTY session.

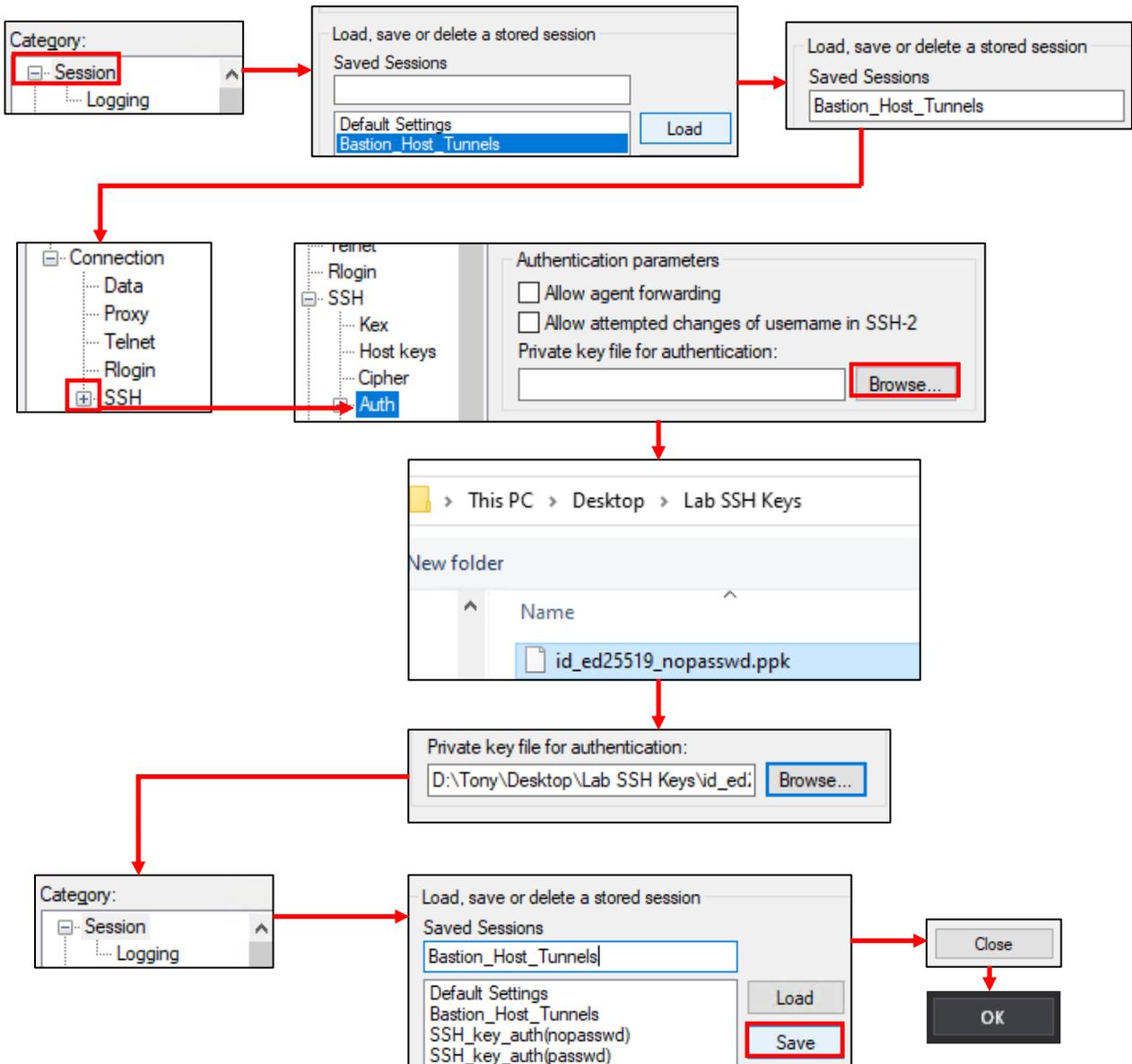
Note: If students want to rename the *Bastion_Host_Tunnels* PuTTY session to something that reflects the fact that key-based authentication has been enabled, Students may modify the name in the *Saved Sessions* input box (e.g., *Bastion_Host_Tunnels(nopasswd)* for passwordless key-based auth, or *Bastion_Host_Tunnels(passwd)* for key-based auth with a password protected private key.), then click *Save*, to save the bastion host PuTTY session.

- Click *Close* to exit the *PuTTYNG Configuration* window, then *OK* to exit the *mRemoteNG Options* window.



16-44: To create a PuTTY session to enable key-based authentication for the SIEM, IPS, and kali VMs, open mRemoteNG, and access the *PuTTYNG Configuration* window found under *Tools > Options*, then in the *mRemoteNG Options* window, *Advanced > Launch PuTTY*. In the *Category* pane, click the "+" next to *SSH*, then click the text *Auth*. To the right of the *Private key file for authentication* input box, click *Browse*, and locate the .ppk generated earlier.

Next, click *Session* in the left pane, and under *Load, save or delete a stored session*, input a name for this session into the *Saved Sessions* input box. Click *Save*, and the new session will be listed in the pane below. Finally, Click *Close* in the *PuTTYNG Configuration* window, then *OK* in *mRemoteNG Options* to exit.



16-45: To edit the existing bastion host PuTTY session, and enable key-based authentication, open the *PuTTYNG Configuration* window once more. Under the *Category* pane, click *Session*, under the *Load, save or delete a stored session* section, click on *Bastion_Host_Tunnels* (or the name students provided to the SSH session of their bastion host), then click the *Load* button. Notice the name of the session appears in the *Saved Sessions* input box.

Next, under the *Category* pane, click the "+" next to *SSH*, then click *Auth* once more. Click the *Browse* button to the right of the *Private key file for authentication* input box once more, and locate the .ppk file generated earlier. When finished, navigate back to *Session*, then click the *Save* button to save these new changes to the *Bastion_Host_Tunnels* PuTTY session. *Close* the PuTTYNG Configuration window, then click *OK* to exit the *mRemoteNG Options* window.

16.5.7: Reconfiguring Connection Profiles, and Testing Key-Based Authentication

The only thing left for students to do is reconfigure the mRemoteNG connection profiles for the SIEM, IPS, and kali virtual machines, then confirm key-based authentication is working properly.

The bastion host's connection profile should not need to be altered. However, since the bastion host is responsible for the SSH forward tunnels that allow connectivity to the lab VMs we're testing, it is still recommended to test key-based authentication and verify the PuTTY session was updated correctly. So that's where we'll start.

Open mRemoteNG, and double click on the bastion host's profile in the *Connections* pane. If students are already connected to the bastion host over SSH, they can right-click on the bastion host's tab, and select *Reconnect*. Upon reconnecting students should see the following lines:

```
Using username "[bastion host username]"
Authenticating with public key "[keyname]"
```

If students opted to not set a password on their SSH private key, they should be connected to their bastion host with no password required. However, if a password was assigned to the private key, students will be prompted it to enter it with the line:

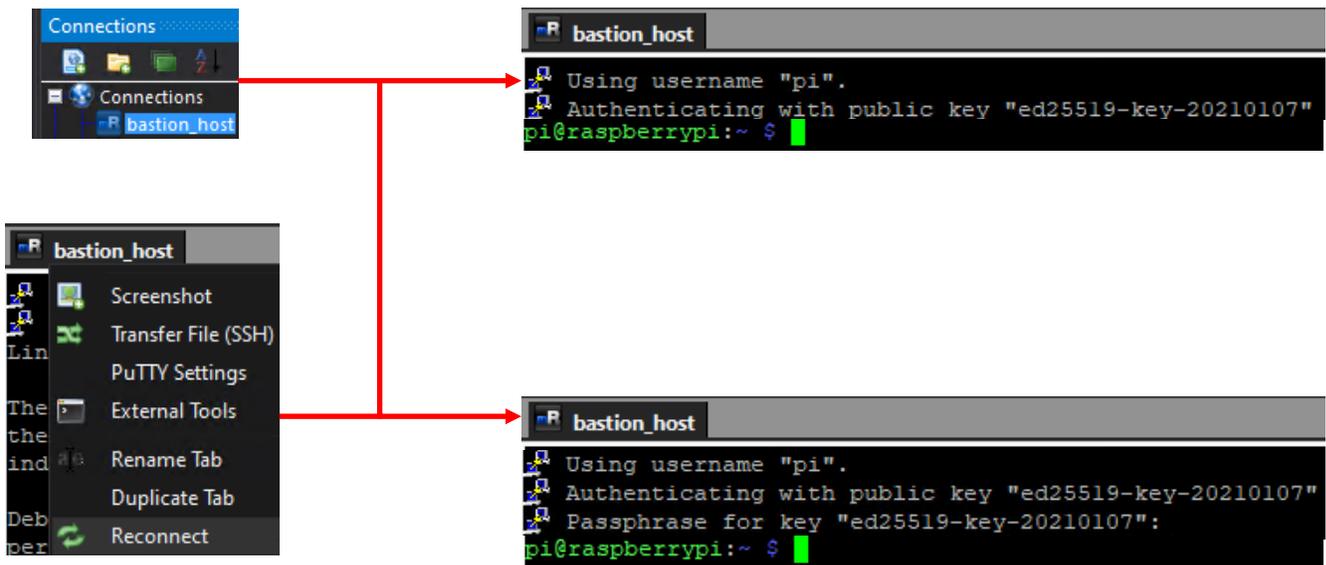
```
Passphrase for key "[keyname]":
```

And after successfully entering the password, will be connected to their bastion host.

Now that students are connected to their bastion host system with both SSH tunneling and key-based authentication enabled, the next step is to reconfigure the connection profiles for the SIEM, IPS, and kali virtual machines. Begin by clicking on the SIEM VM's connection profile in the *Connections* pane, in order to display its configuration in the *Config* pane below it. Under the *Protocol* section, locate the *PuTTY Session* field. Currently it should be set to *Default Settings*. Click the downward facing arrow head to the right, and select the PuTTY session created (e.g., *SSH_key_auth(passwd)*, or *SSH_key_auth(nopasswd)*) in section 16.5.6. **Do not use the Bastion Host Tunnels profile.** Repeat this process for the IPS and kali connection profiles.

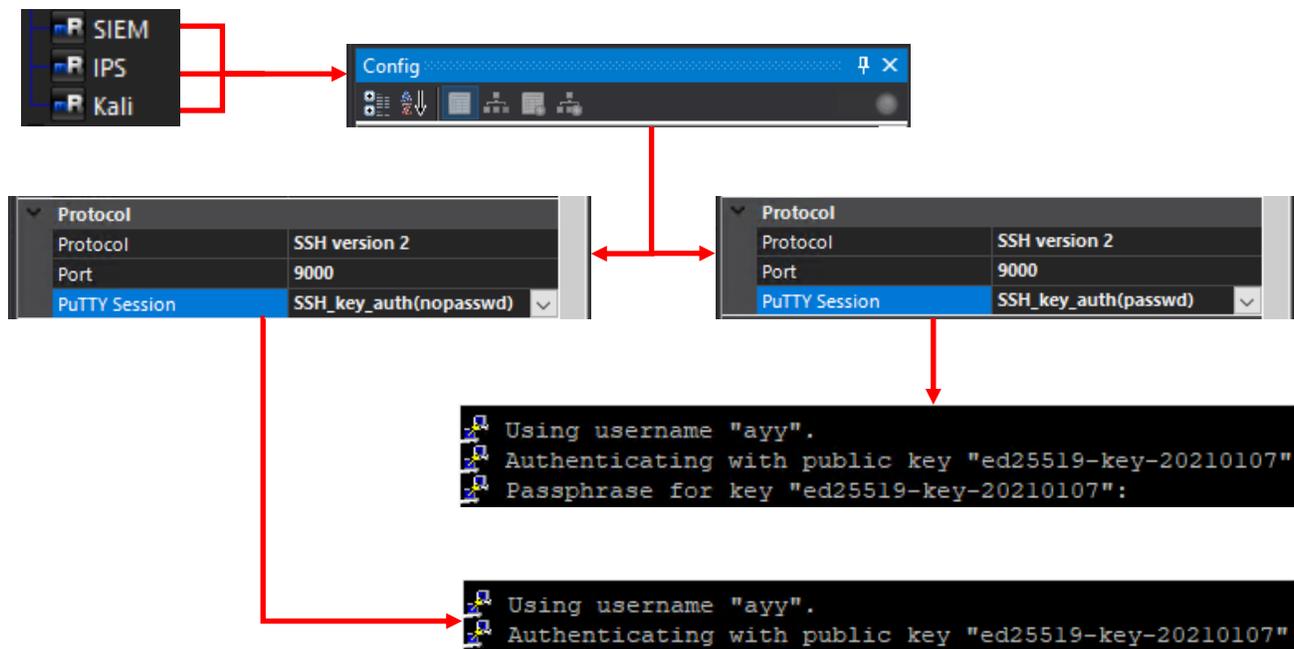
After students have modified all three mRemoteNG connection profiles, double click on each them in the *Connections* pane, to establish SSH sessions (or if students have active sessions, right click on the tab and select *Reconnect*). Just like with the bastion host SSH session, success will either result in an instant SSH session on each host (private key with no password), or will prompt students to enter the private key's password before providing an SSH session on the host (two-factor authentication).

If students are having trouble with key-based authentication or SSH to their lab systems in general, check out [section 16.7](#), starting on page 942.



16-46: To test the changes made to the *Bastion_Host_Tunnels* PuTTY session, and prepare to test configuration changes to the SIEM, IPS, and kali mRemoteNG connection profiles, either double click the bastion host's connection profile under the *Connections* pane to open a new connection, or right click on the bastion host's SSH session tab, and select *Reconnect* to reconnect an active SSH session.

Depending on whether or not students decided to password protect their SSH private key, this will either result in an immediate connection, or a prompt for the SSH private key's password. Entering the correct password for the SSH private key will allow connection to the bastion host, and serve as two-factor authentication.



16-47: The next step is modifying the SIEM, IPS, and kali mRemoteNG connection profiles. Specifically, modifying the PuTTY Session field under the Protocol section for each profile to use the PuTTY session students created in section 16.5.6 for the SIEM, IPS, and kali virtual machines. **Do not use the Bastion Host Tunnels PuTTY session for anything except the bastion host.**

Once the connection profiles have all been reconfigured, connect (or reconnect) to the lab virtual machines over SSH. If everything was done correctly, students will either instantly gain an SSH session on the lab VM (no password protection on private key), or will be prompted to provide the password for the private key (two-factor authentication).

Bonus Lesson: Key-Based Authentication with WinSCP

Way back in Chapter 1, students were advised to have the application WinSCP installed on their system. In fact, one of the (easier) methods for transferring the `authorized_keys` file for SSH key-based authentication had us utilize WinSCP. In the coming chapters (and perhaps, later when and if you choose to customize your lab environment to better suit your needs), you may need to transfer additional files to or from your workstation/hypervisor host to your lab virtual machines. WinSCP will be extremely useful for performing that task quickly and effectively.

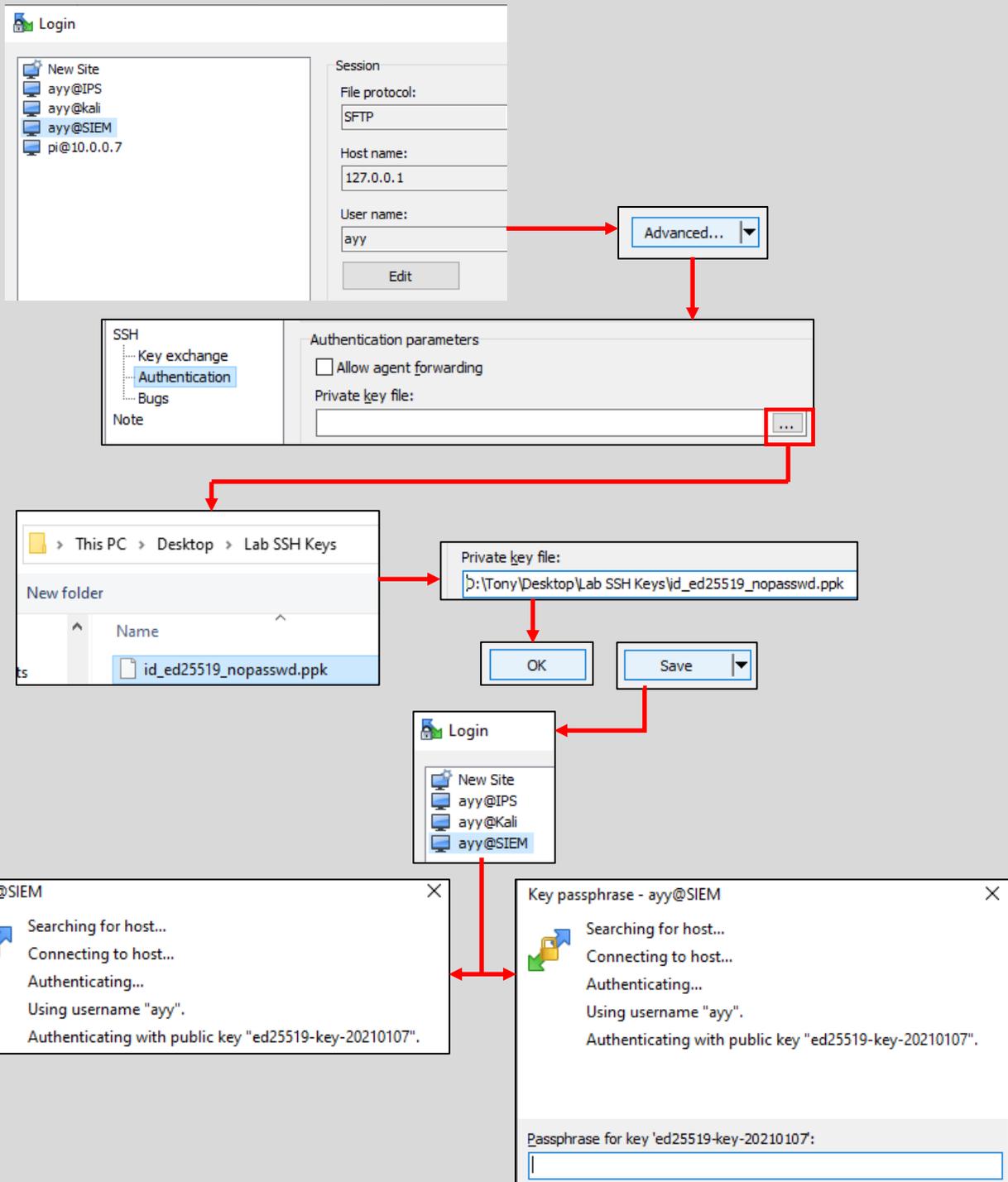
Conveniently, WinSCP supports key-based authentication for its connection profiles, just like mRemoteNG. Also conveniently, it wants SSH private key in `.ppk` format. Please note that the instructions below assume that you have already performed the steps necessary to configure SSH key-based authentication for the SIEM, IPS and Kali virtual machines and that it is working properly.

Open the WinSCP application. If you followed the process for using Method 1 to transfer your `authorized_keys` file to the SIEM, IPS, and Kali VMs, you should already have connection profiles configured for all three VMs. Check out [section 16.5.5.1](#) (pp. 903-906) if you need guidance on how to perform this task. We'll begin by editing the SIEM connection profile. In the *Login* window, select the SIEM connection profile on the left pane, then click the *Edit* button on the right portion of the window, in the *Session* section. Next, click the button labeled *Advanced*, and a new window labeled *Advanced Site Settings* will appear.

In the pane on the left, click on the option labeled *Authentication*, under *SSH*, and a variety of options will appear on the right side of the window. We're interested in the input box labeled *Private key file*, under the *Authentication parameters* section. Click the grey button (labeled ...) on the far right in the input box to open a file browser. Browse to the location of the `.ppk` file we generated earlier using *puttygen*, and double click on it to select it. The input box will update to reflect the full file path to private key. Click the *OK* button below to close the *Advanced Site Settings* window. Back in the *Login* window, click the *Save* button to save the changes we've made to the SIEM connection profile. Repeat this process with the IPS and Kali connection profiles.

When finished, test out your new connection profiles to make sure that key-based authentication is working properly. Open an SSH session to the bastion host to activate the SSH forward tunnels, then Double click on the SIEM connection profile, and confirm you are able to get an SCP session either without a password (if you chose not to password protect your private key file) or with the password used to encrypt your private key file. Again, repeat this process on the bastion host, IPS, and Kali connection profiles to verify key-based authentication is working as intended.

Please be advised that later in this book, students will need to transfer some installer files to some of the lab virtual machines, and that SCP is going to be all but required to do that. **If you decided to disable password authentication over SSH on your lab virtual machines later in this chapter (Section 16.9.2), that WinSCP will no longer work without configuring key-based authentication.**



16-48: Open WinSCP, select the SIEM connection profile, then click *Edit*, followed by *Advanced*. On the *Advanced Site Settings* screen, click on *Authentication*. In the *Private key file* input box, click the grey button with the ellipsis (...) and browse to the location of your .ppk file. When finished, click *OK*, then back in the *Login* window, click *Save*. Repeat this process for the IPS and Kali profiles. Once finished, open an SSH session to the bastion host system (to activate the SSH forward tunnels), then double click on the SIEM connection profile and test to confirm that key-based authentication is working as intended. Depending on whether or not your password protected your private key, you may either be connected automatically, or prompted for the password to decrypt your private key, if everything is working properly. Configure and test the bastion host, IPS, and Kali site profiles as well.

16.6 Establishing SSH Connectivity to the Bastion Host and Lab VMs (Linux/MacOS)

In this section, I will instruct students on how to configure SSH connections to their bare-metal hypervisor lab on Linux or MacOS using the `ssh` command. We will then create a `~/.ssh/config` file that we can use to make connecting to our lab hosts much easier. We will first establish SSH connectivity to the bastion host, then configure forward tunnels to the SIEM, IPS, and Kali virtual machines, as well as a dynamic tunnel for later use. After connectivity to the bastion host and lab virtual machines has been established, students will be guided through the process on how to generate an SSH public/private key pair, as well as how to enable key-based authentication to their lab virtual machines, and bastion host. **Windows users should jump back to section 16.5, starting on page 889.**

16.6.1 The `ssh` command

Connecting to hosts via the SSH protocol is a little bit easier on Linux/MacOS because the `ssh` command, along with a couple of other handy utilities that makes setup a little more convenient. As described in Chapter 15, [section 15.3.1 \(pp. 784-787\)](#), the `ssh` command is should be easy to use:

```
ssh [username]@[Hostname/IP address]
```

If the local workstation is able to figure out how to reach the remote system, and the connection to the service is accepted, the user will be prompted to enter a password. If the correct password is supplied, the user gets command-line access to the remote system. Students should open a their terminal application, and try using the `ssh` command to connect to their bastion host system. The name of the user on my bastion host is "ayy", and its IP address is "10.0.0.162". So, my `ssh` command would look like:

```
ssh ayy@10.0.0.162
```

Please note that upon first connection, the `ssh` client may prompt students about the authenticity of the SSH host key, asking if they wish to continue connecting. To make a long story short, respond *yes* to continue, then enter the password for the bastion host user when prompted. This should result in successful SSH connectivity to the bastion host. Type `exit` to close the SSH session.

Note: If you're really unlucky and mess with virtual machines and/or have systems that re-use IP addresses on your local network a lot, you might get very nasty message about how the remote host's identification has changed. The nastygram will go on about how you're possibly being targeted by a man in the middle attack and then refuse to connect to the remote system. If this happens to you, or you want more details about SSH host keys, check out the sidebar conversation in section 15.3.1 [SSH Host Keys, but MacOS/Linux \(pp. 786-787\)](#) for more information, and a solution to this problem.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ayy@10.0.0.162
The authenticity of host '10.0.0.162 (10.0.0.162)' can't be established.
ECDSA key fingerprint is SHA256:Xf7uyRGVHshpWaCgvQFpJi00i5zzegAWaQ74J3Nrj5g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.162' (ECDSA) to the list of known hosts.
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)

ayy@bastion:~$ whoami; hostname; ip -br addr
ayy
bastion
lo                UNKNOWN          127.0.0.1/8  :
ens160            UP               10.0.0.162/24
ayy@bastion:~$ exit
logout
Connection to 10.0.0.162 closed.
```

16-49: Using the ssh command is pretty easy. Using a terminal application, run the command:

```
ssh [username]@[Hostname/IP address]
```

If prompted to save the SSH host key/fingerprint, select yes. Enter the remote user's password when prompted. If its correct, students will get shell access to their bastion host. Type exit to disconnect from the remote system.

16.6.2 Enabling and Testing SSH tunnels

Connecting to a remote system via the ssh command is easy enough, but how do we enable the necessary forward and dynamic tunnels in order to access the rest of the lab environment? The ssh command has a variety of different options or "switches" that can be used to change how the client operates. A few of these options include the -L, the -R, and the -D switches. The -L option adds a local listener (forward tunnel), while -R adds a remote listener (reverse tunnel). That leaves the -D option, used for dynamic tunnels.

As far as the lab environment goes right now, students will need to craft an SSH tunnel with three -L tunnels, one for each of the lab virtual machines, and one -D tunnel for later use as a web proxy:

```
ssh -L9000:[SIEM VM IP]:22 -L9001:[IPS VM IP]:22 -L9002:[kali VM IP]:22 -D9003
[username]@[bastion host IP]
```

For example, in my lab environment, this command would look like this:

```
ssh -L9000:172.16.1.3:22 -L9001:172.16.1.4:22 -L9002:172.16.2.2:22 -D9003
ayy@10.0.0.162
```

If students entered the commands correctly, they should be prompted with the password of their bastion host user. When entered, an SSH session should be established to the bastion host as normal.

Note: If students are using an alternate IP address scheme for their lab environment, replace the IP addresses 172.16.1.3, 172.16.1.4, and 172.16.2.2 with the IP addresses of the SIEM, IPS, and kali virtual machines, respectively. By the way, for the SSH tunnels we'll be using for the lab, you don't have to use ports 9000 through 9003 if you don't want to. The TCP ports you pick also don't have to be sequential. However, for SSH tunnels, I recommend picking any TCP ports between port 9000, and 65535.

The SSH forward tunnels are supposed to enable SSH connectivity to the SIEM, IPS, and kali virtual machines by way of our bastion host, so the next step is testing this connectivity to see if it's working as intended. **For this next step, do not disconnect the SSH session to the bastion host system.** Students should open new terminal sessions, or new tabs in their terminal sessions and run the following commands:

```
ssh -p 9000 [SIEM username]@127.0.0.1
ssh -p 9001 [IPS username]@127.0.0.1
ssh -p 9002 [kali username]@127.0.0.1
```

Notice that all three of these SSH sessions are for the IP address 127.0.0.1 (localhost), but all of them have the `-p` option in the ssh command. This option allows users to specify what TCP port number to connect to on the remote host. Each of these commands is telling the SSH client to connect to the remote host, 127.0.0.1 (localhost) on port 9000, 9001, and 9002. These port numbers correspond with the port numbers we used for the three `-L` (forward tunnels) we configured for the SSH session to the bastion host. TCP connections on those ports, are being relayed through the bastion host, who has static routes to the lab networks, and firewall rules that allow it to access all of those hosts on port 22/TCP (the default port for the SSH service).

If students have done everything correctly up to this point, the end result should be an SSH session on the SIEM, IPS and kali virtual machines, by way of the forward tunnels through the bastion host system.

```
ssh -L9000:172.16.1.3:22 -L9001:172.16.1.4:22 -L9002:172.16.2.2:22 -D9003 ayy@10.0.0.162
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
-L9000:172.16.1.3:22
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh -p 9000 ayy@127.0.0.1
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$ whoami; hostname; ip -br addr
ayy
siem
lo                UNKNOWN          127.0.0.1/8 :
ens160            UP               172.16.1.3/24
```

```
-L9001:172.16.1.4:22
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh -p 9001 ayy@127.0.0.1
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$ whoami; hostname; ip -br addr
ayy
ips
lo                UNKNOWN          127.0.0.1/8 :
ens160            UP               172.16.1.4/24
ens192            DOWN
ens224            DOWN
```

```
-L9002:172.16.2.2:22
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh -p 9002 ayy@127.0.0.1
ayy@127.0.0.1's password:
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@Kali:~$ whoami; hostname; ip -br addr
ayy
Kali
lo                UNKNOWN          127.0.0.1/8 :
eth0              UP               172.16.2.2/24
```

16-50: In order to create SSH tunnels, and test connectivity to the lab VMs through them, run the following command:

```
ssh -L9000:[SIEM VM IP]:22 -L9001:[IPS VM IP]:22 -L9002:[kali VM IP]:22 -D9003
[username]@[bastion host IP]
```

After successfully connecting to the bastion host VM, run the following commands (in separate terminal sessions/tabs) to test connectivity to the SIEM, IPS, and kali VMs:

```
ssh -p 9000 [SIEM username]@127.0.0.1
ssh -p 9001 [IPS username]@127.0.0.1
ssh -p 9002 [kali username]@127.0.0.1
```

Notice how the port numbers of the -L options (9000-9002) in the bastion host ssh command correspond to the port numbers of the -p option in the ssh commands to connect to the SIEM, IPS, and kali VMs.

Checking your tunnels with netstat and ss

Aside from just attempting to SSH to the lab virtual machines, another way to see if the forward and dynamic tunnels are up and accepting connections are through the macOS `netstat` command, and the Linux `ss` command.

If you're on a macOS host, run the following:

```
netstat -an -p tcp -f inet | grep LISTEN | grep 900
```

The `netstat` command is very powerful and will show practically all current network activity on the system. The arguments supplied to the `netstat` command above tell it to show us all connections (`-a`), numeric output only (`-n`), only display TCP connections (`-p tcp`), and only IPv4 connections (`-f inet`). When we get the output from `netstat` we send it through a pipe (`|`) to the `grep` command. `grep` is a powerful tool that can be used to search output from files and commands and search only for particular output you're interested in. In this case, we're telling `grep` that we're interested in lines that contain the string "LISTEN". Afterwards, we pipe the output from that into another `grep` command. Out of the output from the previous command, only show us lines that contain the string "900". Remember how we set up SSH tunnels on ports 9000 through 9003? This `grep` command will match on all of those ports, since the numbers "900" is a substring of 9000, 9001, 9002, and 9003. If the SSH tunnels were set up correctly, there should be four lines of output that look something like this:

```
tcp4      0      0 127.0.0.1.9003      *.*          LISTEN
tcp4      0      0 127.0.0.1.9002      *.*          LISTEN
tcp4      0      0 127.0.0.1.9001      *.*          LISTEN
tcp4      0      0 127.0.0.1.9000      *.*          LISTEN
```

These are our three forward tunnels, and single dynamic tunnel, listening on the localhost address, waiting for connections.

Now, for Linux users, you get to use the `ss` command. Try this out:

```
ss -a1nt4 | grep 900
```

The `ss` command (shorthand for socket statistics) is, more or less, a re-write of `netstat`. Technically, the `netstat` command *is* available for Linux (and its wildly different from the `netstat` command on Windows or MacOS), but `ss` is considered the "new and improved" `netstat`, and is usually available by default on most modern Linux distributions. With all that out of the way, what did we just do?

We ran `ss`, and told it to show us all connections (`-a`), show us sockets listening for connections only (`-l`), provide numeric output only (`-n`), show us TCP (`-t`) and IPv4 connections only (`-4`). We then pipe the output from that command to `grep`, and search for the string `900`. Once again, if the tunnels are up and waiting for connections, students should get four lines that look something like this:

```
LISTEN 0      128          127.0.0.1:9000      0.0.0.0:*
LISTEN 0      128          127.0.0.1:9001      0.0.0.0:*
LISTEN 0      128          127.0.0.1:9002      0.0.0.0:*
LISTEN 0      128          127.0.0.1:9003      0.0.0.0:*
```

(Author's note: Chances are, if you're reading this, and prefer Linux `netstat` over `ss`, you already know how to install it, and run it to get the output we're looking for. Yerr a network wizard, 'arry.)

16.6.3 Creating SSH connection profiles via `~/.ssh/config`

Now that students have had a chance to confirm SSH connectivity to their bastion host, as well as the SIEM, IPS, and kali VMs, the next subject we'll cover is creating connection profiles for those systems using `~/.ssh/config`. To make a very long story short, this configuration file can store connection preferences that the `ssh` command will "remember" when connection to certain hosts. It's even possible to set names for the hosts. For example, students could run `ssh bastion_host`, and with a properly formatted config file, it would know the correct IP address to connect to, the proper user to log in as, and all of the necessary SSH tunnels to spawn on a successful connection.

Like most of the other subjects in this chapter, the format for the `~/.ssh/config` file was covered in chapter 15 – specifically, [section 15.3.2](#) (pp. 788-797). In order to save time (and pages, and trees), students will be provided with a bulleted list of tasks to complete, based on the instructions from chapter 15, adapted with additional options needed for bare-metal hypervisors.

- Students should copy the sample SSH config below to the Linux/macOS workstation, using their favorite text editor, and modify the fields to reflect their lab environment:

SSH Config

```
Host bastion_host
    Hostname 10.0.0.162
    User ayy
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003
```

```
Host siem
    Hostname 127.0.0.1
    Port 9000
    User ayy
```

```
Host ips
    Hostname 127.0.0.1
    Port 9001
    User ayy
```

```
Host kali
    Hostname 127.0.0.1
    Port 9002
    User ayy
```

- A copy of this file is available at:
<https://gist.github.com/da667/652ebe408b65d5cc8a289a4889c46798>

Note: Students may need to adjust the `Hostname` field to reflect the IP address of their lab systems (e.g., the bastion host, or the SIEM, IPS and kali VMs). They may also need to adjust the `User` field to reflect the name of the user they want to use to log in to these systems. Finally, the TCP port number specified in the `Port` field for the SIEM, IPS and kali entries must match the port numbers used in the `LocalForward` directives for the `bastion_host` entry.

- With the file copied/downloaded, open a terminal session, and use `cd` command to change directories to the location of the newly created config file. First, run the following command:

```
ls -al ~/.ssh
```

- If the command returns the following output:

```
ls: cannot access '/home/[username]/.ssh': No such file or directory
```

run these commands:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
cp config ~/.ssh
chmod 600 ~/.ssh/config
```

- If the `config` file already exists in `~/.ssh`, run:

```
cp ~/.ssh/config ~/.ssh/config.old
cat config >> ~/.ssh/config
```

Note: If you already have a `~/.ssh/config` file on your workstation, and you run into problems trying to connect to your lab systems, check out the sidebar conversation in Section 15.3.2, [*What if I'm already using a ~/.ssh/config?*](#) (page 793)

- Otherwise, if the `~/.ssh` directory exists, and the `config` file does NOT exist, run:

```
cp config ~/.ssh
chmod 600 ~/.ssh/config
```

- With the `config` file in the correct location with the correct file permissions run the following commands:

```
ssh bastion_host
ssh siem
ssh ips
ssh kali
```

- Remember that the SSH connection to the bastion host must be kept active for the SSH forward tunnels to operate correctly

```
trobinson@trobinsons-MacBook-Pro ~ % vi ~/config
trobinson@trobinsons-MacBook-Pro ~ % cat ~/config
Host bastion_host
    Hostname 10.0.0.162
    User ayy
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003

Host siem
    Hostname 127.0.0.1
    Port 9000
    User ayy

Host ips
    Hostname 127.0.0.1
    Port 9001
    User ayy

Host kali
    Hostname 127.0.0.1
    Port 9002
    User ayy

trobinson@trobinsons-MacBook-Pro ~ % ls -al ~/.ssh
total 16
drwx----- 3 trobinson  staff   96 Mar 29 14:13 .
drwxr-xr-x+ 31 trobinson  staff  992 Mar 29 14:15 ..
-rw----- 1 root        staff 5621 Mar 29 13:50 known_hosts
trobinson@trobinsons-MacBook-Pro ~ % cp ~/config ~/.ssh
trobinson@trobinsons-MacBook-Pro ~ % chmod 600 ~/.ssh/config
```

Continued to *fig. 16-52*

16-51: Begin by creating the SSH config file, using the illustration above as an example. Students should change the User, Port, and Hostname fields to reflect their lab environment as needed. The ~/.ssh directory should already exist if students testing SSH connectivity to their lab virtual machines, and bastion host. Use the cp command to copy the config file to ~/.ssh, followed by chmod 600 ~/.ssh/config to set the correct file permissions.

Continued from *fig. 16-51*

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$ whoami; hostname; ip -br addr
ayy
siem
lo                UNKNOWN        127.0.0.1/8 :
ens160            UP              172.16.1.3/24
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ips
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$ whoami; hostname; ip -br addr
ayy
ips
lo                UNKNOWN        127.0.0.1/8 :
ens160            UP              172.16.1.4/24
ens192            DOWN
ens224            DOWN
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh kali
ayy@127.0.0.1's password:
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@Kali:~$ whoami; hostname; ip -br addr
ayy
Kali
lo                UNKNOWN        127.0.0.1/8 :
eth0              UP              172.16.2.2/24
```

16-52: With the SSH config file in place, begin by running the command:

```
ssh bastion_host
```

If the SSH config file was created properly, this command will attempt to connect students to their bastion host system. Enter the bastion host user's password to log in. Once logged in, create a new terminal tab/session and try to connect to the SIEM, IPS, and kali VMs with the following commands:

```
ssh siem
ssh ips
ssh kali
```

Once successful login has been confirmed, students may type exit to disconnect.

Aliases: An Alternative to ~/.ssh/config

If students don't want to muck about creating a ~/.ssh/config file, an alternative involves using the alias command. In a nutshell, alias allows you to define a string or a word as a complex command or a complex series of commands. Many Unix/Linux-based operating systems have a host of aliases defined by default, and these can be viewed by running alias with no arguments. The only problem is that aliases by default, are not persistent. However, they can be made persistent by storing them in a configuration file used by your command-line environment. I covered the alias command in chapter 15, section 15.3.2 (pp. 788-797), the sidebar conversation *Alternative Method: The alias Command*. Here is a list of actions to perform, if you'd rather create SSH connection strings using the alias command instead:

- Open your Linux or MacOS workstation's terminal application, and run the command:

```
echo $SHELL
```

- In most cases, this command should return the output /bin/zsh indicating that the z shell environment is used, or /bin/bash, indicating that the bash shell environment is used.

- If the bash shell environment is being used, run the following commands:

```
echo "alias bastion_host='ssh -L9000:[SIEM VM IP]:22 -L9001:[IPS VM IP]:22  
-L9002:[kali VM IP]:22 -D9003 [username]@[bastion host IP]'" >> ~/.bashrc  
echo "alias siem='ssh -p 9000 [SIEM username]@127.0.0.1'" >> ~/.bashrc  
echo "alias ips='ssh -p 9001 [IPS username]@127.0.0.1'" >> ~/.bashrc  
echo "alias kali='ssh -p 9002 [kali username]@127.0.0.1'" >> ~/.bashrc
```

- If the zsh shell environment is being used, run the following, instead:

```
echo "alias bastion_host='ssh -L9000:[SIEM VM IP]:22 -L9001:[IPS VM IP]:22  
-L9002:[kali VM IP]:22 -D9003 [username]@[bastion host IP]'" >> ~/.zshrc  
echo "alias siem='ssh -p 9000 [SIEM username]@127.0.0.1'" >> ~/.zshrc  
echo "alias ips='ssh -p 9001 [IPS username]@127.0.0.1'" >> ~/.zshrc  
echo "alias kali='ssh -p 9002 [kali username]@127.0.0.1'" >> ~/.zshrc
```

- For both of the strings of commands above, be sure to substitute IP addresses and usernames and port numbers you used for your forward tunnels as necessary for your lab environment. For example these are the aliases I created for my lab environment:

```
echo "alias bastion_host='ssh -L9000:172.16.13:22 -L9001:172.16.1.4:22  
-L9002:172.16.2.2:22 -D9003 ayy@10.0.0.162'" >> ~/.zshrc  
echo "alias siem='ssh -p 9000 ayy@127.0.0.1'" >> ~/.zshrc  
echo "alias ips='ssh -p 9001 ayy@127.0.0.1'" >> ~/.zshrc  
echo "alias kali='ssh -p 9002 ayy@127.0.0.1'" >> ~/.zshrc
```

- When finished, the aliases need to be tested. Open the terminal application on your Linux or MacOS workstation and begin by typing `bastion_host` and hitting enter. If you're prompted to enter a password, for the bastion host's user, the alias worked. Enter the user's password, and set that SSH session aside, because you'll need the SSH forward tunnels from that session to test the other aliases
- With the SSH session to the bastion host active, open a new terminal session or tab and run the other aliases:

```
siem
ips
kali
```

- Confirm that these aliases result in SSH sessions to the SIEM, IPS, and kali VMs, respectively

Bear in mind that future sections in this chapter assume you created entries in the `~/.ssh/config` file for the bastion host and lab VMs. You may need to read up on the documentation of the `ssh-copy-id`, `scp`, or other SSH-related commands to know how to specify port numbers, usernames, etc.

```
trobinson@trobinsons-MacBook-Pro .ssh % echo $SHELL
/bin/zsh
trobinson@trobinsons-MacBook-Pro .ssh % echo "alias bastion_host='ssh -L9000:172.16.1.3:22 -L9001:172.16.1.4:22 -L9002:172.16.2.2:22 -D9003 ayy@10.0.0.162'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro .ssh % echo "alias siem='ssh -p 9000 ayy@127.0.0.1'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro .ssh % echo "alias ips='ssh -p 9001 ayy@127.0.0.1'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro .ssh % echo "alias kali='ssh -p 9002 ayy@127.0.0.1'" >> ~/.zshrc
trobinson@trobinsons-MacBook-Pro .ssh % cat ~/.zshrc
test -e "${HOME}/.iterm2_shell_integration.zsh" && source "${HOME}/.iterm2_shell_integration.zsh"
alias keepalive='while true; do w; sleep 10; done'

alias bastion_host='ssh -L9000:172.16.1.3:22 -L9001:172.16.1.4:22 -L9002:172.16.2.2:22 -D9003 ayy@10.0.0.162'
alias siem='ssh -p 9000 ayy@127.0.0.1'
alias ips='ssh -p 9001 ayy@127.0.0.1'
alias kali='ssh -p 9002 ayy@127.0.0.1'
trobinson@trobinsons-MacBook-Pro ~ % bastion_host
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

16-53: To add SSH connection aliases, begin by running:

```
echo $SHELL
```

If the shell is `bash`, the aliases should be stored in `~/.bashrc`. If the shell is `zsh`, store them in `~/.zshrc`. Next, students will need to copy 4 aliases into the appropriate shell source file. Substitute IP addresses, port numbers, and names of user accounts on the different systems as necessary. Finally, open a new terminal window and run the alias `bastion_host`. If the alias results in an SSH connection attempt to the bastion host system, log in and keep that SSH session active to activate the forward tunnels for the other lab VMs.

```
trobinson@trobinsons-MacBook-Pro ~ % siem
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$ whoami; hostname; ip -br addr
ayy
siem
lo                UNKNOWN        127.0.0.1/8 :
ens160            UP             172.16.1.3/24
```

```
trobinson@trobinsons-MacBook-Pro ~ % ips
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$ whoami; hostname; ip -br addr
ayy
ips
lo                UNKNOWN        127.0.0.1/8 :
ens160            UP             172.16.1.4/24
ens192            DOWN
ens224            DOWN
```

```
trobinson@trobinsons-MacBook-Pro ~ % kali
ayy@127.0.0.1's password:
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@Kali:~$ whoami; hostname; ip -br addr
ayy
Kali
lo                UNKNOWN        127.0.0.1/8 :
eth0              UP             172.16.2.2/24
```

16-54: With an SSH session to the bastion host active, open a new terminal window or tab, and test the siem, ips, and kali aliases and confirm that each alias provides an SSH connection to the correct virtual machine.

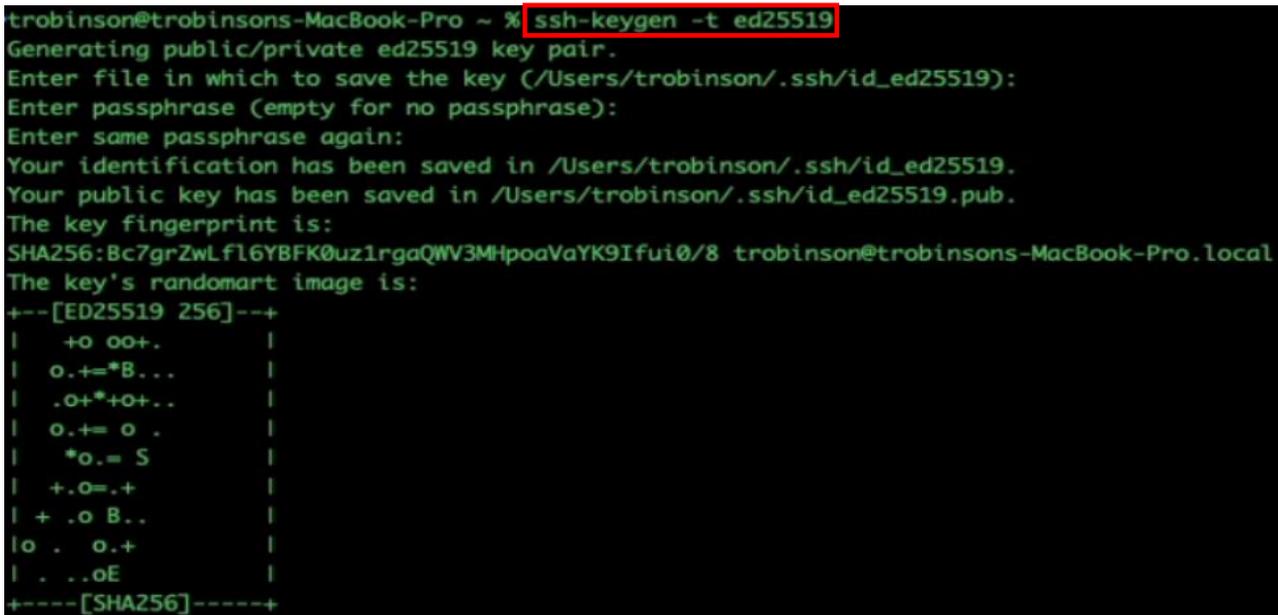
16.6.4 Generating SSH Keys for Key-Based Authentication (Optional)

In this section students will learn how to generate a public/private key pair in order to enable SSH key-based authentication to their bastion host, as well as the SIEM, IPS, and kali virtual machines. As the name of this section implies, this content is entirely optional. As with most of the other content in this chapter, this content was covered in Chapter 15, specifically [section 15.3.3.1](#) (pp. 798-799). Here is a list of tasks to perform in order to generate an SSH key:

- Open the terminal application on the Linux or MacOS workstation and run the command:

```
ssh-keygen -t ed25519
```

- Students will be prompted where to save the SSH keys this command. Hit enter to accept the default location (`~/.ssh/id_ed25519`)
- Students will then be prompted twice to enter a passphrase for the private key
- If students want to enable passwordless SSH authentication, press enter twice to skip assigning a password to the private key
- If students wish to enable two-factor authentication for SSH sessions, enter the same password twice, and save that password to a password manager for future use



```
trobinsont@trobinsonts-MacBook-Pro ~ % ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/trobinsont/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/trobinsont/.ssh/id_ed25519.
Your public key has been saved in /Users/trobinsont/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Bc7grZwLfl6YBFK0uz1rgaQWV3MHpoaVaYK9Ifui0/8 trobinsont@trobinsonts-MacBook-Pro.local
The key's randomart image is:
+--[ED25519 256]--+
|  +o oo+. |
| o.+=*B... |
| .o+*+o+.. |
| o.+ o . |
| *o.= S |
| +.o=.+ |
| + .o B.. |
| o . o.+ |
| . ..oE |
+-----[SHA256]-----+
```

16-55: To generate SSH keys on Linux/MacOS, run:

```
ssh-keygen -t ed25519
```

When asked where to store the SSH private key, press enter to accept the default location (`~/.ssh/id_ed25519`). If students want to enable passwordless SSH authentication, press enter twice to skip entering a password for the SSH private key. Otherwise, assign a password to the private key (and save that password) to enable two-factor authentication for key-based SSH sessions.

16.6.5 Copying The `authorized_keys` File to the Bastion Host, and Lab VMs

Students should now have a key pair generated by `ssh-keygen: id_ed25519` – the private key, and `id_ed25519.pub` – the public key. The next step students need to perform in order to enable key-based authentication involves transferring the public key file, to the SIEM, IPS, and Kali VMs, as well as the bastion host. In this section, students will be provided with a bulleted list of tasks to perform in order to transfer the public key file to the bastion host and lab virtual machines, using any one of these three methods:

1. Utilizing `ssh-copy-id`
2. The `scp` command
3. Copy, paste, and output redirection

As with most of the content covered in this chapter, These transfer methods were also covered in chapter 15. If students need more details about how the different methods work, check out [section 15.3.3.2 \(pp. 799-807\)](#)

16.6.5.1 Method 1: `ssh-copy-id`

- Establish an SSH session to the bastion host system in order to activate the SSH forward tunnels that will allow access to the SIEM, IPS, and Kali VMs, using the command:

```
ssh bastion_host
```

- With an SSH session established to the bastion host, run the following commands (in separate terminal sessions or tabs):

```
ssh-copy-id bastion_host
ssh-copy-id siem
ssh-copy-id ips
ssh-copy-id kali
```

Note: when running `ssh-copy-id bastion_host`, students may get a collection of error messages:

```
bind [::1]:9000: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9000
bind [::1]:9001: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9001
bind [::1]:9002: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9002
bind [::1]:9003: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9003
Could not request local forwarding.
```

Disregard these error messages. We're only interested in the confirmation that the SSH public key was copied successfully.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh-copy-id bastion_host
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobinson/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
ayy@10.0.0.162's password:
bind [::1]:9000: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9000
bind [::1]:9001: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9001
bind [::1]:9002: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9002
bind [::1]:9003: Address already in use
channel_setup_fwd_listener_tcpip: cannot listen to port: 9003
Could not request local forwarding.

Number of key(s) added:      1

Now try logging into the machine, with: "ssh 'bastion_host'"
and check to make sure that only the key(s) you wanted were added.
```

Continued to *fig. 16-57*

16-56: The first method students can use to copy their SSH public key to their bastion host and lab VMs is the command `ssh-copy-id`. Begin by establishing an SSH session to the bastion host VM using the command:

```
ssh bastion_host
```

Enter the bastion host user's password to get an SSH session. Afterwards, open another terminal session/tab and run the command:

```
ssh-copy-id bastion_host
```

When prompted, enter the bastion host user's password. Students may notice a number of errors:

```
Cannot listen to port: xxxx
Could not request local forwarding
```

Disregard these. The only output students should concern themselves with is

```
Number of key(s) added:      1
```

Continued from *fig. 16-56*

```
trobinsont@trobinsonts-MacBook-Pro ~ % ssh-copy-id siem
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobinsont/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
ayy@127.0.0.1's password:

Number of key(s) added:      1

Now try logging into the machine, with: "ssh 'siem'"
and check to make sure that only the key(s) you wanted were added.
```

```
trobinsont@trobinsonts-MacBook-Pro ~ % ssh-copy-id ips
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobinsont/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
ayy@127.0.0.1's password:

Number of key(s) added:      1

Now try logging into the machine, with: "ssh 'ips'"
and check to make sure that only the key(s) you wanted were added.
```

```
trobinsont@trobinsonts-MacBook-Pro ~ % ssh-copy-id kali
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/trobinsont/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
ayy@127.0.0.1's password:

Number of key(s) added:      1

Now try logging into the machine, with: "ssh 'kali'"
and check to make sure that only the key(s) you wanted were added.
```

16-57: With the SSH public key copied to the bastion host, and with an active SSH session to the bastion host system, run the following commands:

```
ssh-copy-id siem
ssh-copy-id ips
ssh-copy-id kali
```

Enter the password for the user on the SIEM, IPS, and kali virtual machines respectively, and watch for the output:

```
Number of key(s) added:      1
```

16.6.5.2 Method 2: scp

- Begin by establishing an SSH session to the bastion host system, as well as SIEM, IPS, and kali virtual machines.
- Open a new terminal window/tab and run the following commands:

```
scp ~/.ssh/id_ed25519.pub bastion_host:~/authorized_keys
scp ~/.ssh/id_ed25519.pub siem:~/authorized_keys
scp ~/.ssh/id_ed25519.pub ips:~/authorized_keys
scp ~/.ssh/id_ed25519.pub kali:~/authorized_keys
```

- Run the following commands in each of the SSH sessions established to the bastion host, SIEM, IPS, and kali systems:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
mv ~/authorized_keys ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ips
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh kali
ayy@127.0.0.1's password:
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@Kali:~$
```

Continued to *fig. 16-59*

16-58: Begin by establishing SSH sessions to the bastion host, SIEM, IPS, and kali systems.

Continued from *fig. 16-58*

```
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub bastion_host:~/authorized_keys
ayy@10.0.0.162's password:
id_ed25519.pub                               100% 120    18.4KB/s   00:00
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub siem:~/authorized_keys
ayy@127.0.0.1's password:
id_ed25519.pub                               100% 120    16.4KB/s   00:00
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub ips:~/authorized_keys
ayy@127.0.0.1's password:
id_ed25519.pub                               100% 120    18.8KB/s   00:00
trobinson@trobinsons-MacBook-Pro ~ % scp ~/.ssh/id_ed25519.pub kali:~/authorized_keys
ayy@127.0.0.1's password:
id_ed25519.pub                               100% 120    21.3KB/s   00:00
```



```
~$ mkdir ~/.ssh
~$ chmod 700 ~/.ssh
~$ mv ~/authorized_keys ~/.ssh
~$ chmod 600 ~/.ssh/authorized_keys
```

16-59: Run the following commands to transfer the public key file to the bastion host and lab VMs:

```
scp ~/.ssh/id_ed25519.pub bastion_host:~/authorized_keys
scp ~/.ssh/id_ed25519.pub siem:~/authorized_keys
scp ~/.ssh/id_ed25519.pub ips:~/authorized_keys
scp ~/.ssh/id_ed25519.pub kali:~/authorized_keys
```

After transferring the public key file (renaming it to `authorized_keys` in the process) to the SIEM, IPS and Kali VMs using the `scp` command, students will then need to run a series of commands on the bastion host, and all three virtual machines to ensure the `authorized_keys` file is in the correct location with the proper file permissions configured:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
mv ~/authorized_keys ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

Just to re-iterate, these commands must be run on the bastion host as well as the SIEM, IPS and kali virtual machines.

16.6.5.3 Method 3: Copy, Paste, and Output Redirection

- Open a terminal session on your Linux or MacOS workstation, and run the following command:

```
cat ~/.ssh/id_ed25519.pub
```

- The `cat` command is used to print the contents of a text file to the terminal window. This command should have produced a line of text. Highlight this line of text. The line will begin with `ssh-ed25519` and will end with `[username]@[hostname]` (e.g., on my macbook, this is `trobinson@trobinsons-MacBook-Pro.local`) then right click on it and select *Copy*
- Open SSH sessions on the bastion host, followed by the SIEM, IPS, and kali VMs. Run the following commands on each SSH session:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
echo "[right click -> Paste]" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

- In the `echo` command above, students should be pasting the contents of the `id_ed25519.pub` file. As an example, below are the contents of my public key file, and what the command looks like based on the contents of my public key file:

```
echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDYxYwxXvU1qSsQUcUr7AUNz74P2L1npX5dVSMgRrUCI
trobinson@trobinsons-MacBook-Pro.local" >> ~/.ssh/authorized_keys
```

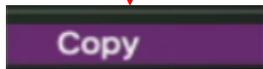
- Consider running the command:

```
wc -l ~/.ssh/authorized_keys.
```

- If the public key was copied correctly, the output should look something like this:

```
1 /home/[username]/.ssh/authorized_keys
```

```
trobinson@trobinsons-MacBook-Pro ~ % cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDYxYwxXvU1qSsQUcUr7AUNz74P2L1npX5dVSMgRrUCI trobinson@trobinson
s-MacBook-Pro.local
```



```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@10.0.0.162's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ips
ayy@127.0.0.1's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh kali
ayy@127.0.0.1's password:
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@kali:~$
```

```
~$ chmod 700 ~/.ssh
~$ echo "ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDYxYwxXvU1qSsQUcUr7AUNz74P2L1npX5dVSMgRrUCI
n@trobinsons-MacBook-Pro.local" >> ~/.ssh/authorized_keys
~$ chmod 600 ~/.ssh/authorized_keys
~$ wc -l ~/.ssh/authorized_keys
1 /home/ayy/.ssh/authorized_keys
```

16-60: In a terminal session on students Linux or MacOS workstation, run:

```
cat ~/.ssh/id_ed25519.pub
```

Copy the output from that command, then open up SSH sessions to the bastion host, followed by the SIEM, IPS and kali virtual machines. Run the following commands in each SSH session:

```
mkdir ~/.ssh
chmod 700 ~/.ssh
echo "[right click -> Paste]" >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
wc -l ~/.ssh/authorized_keys
```

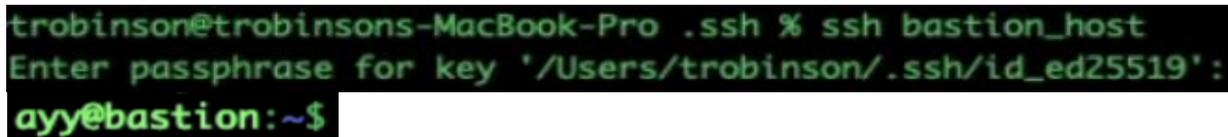
The output of this command should return 1 as the very first character, indicating that the file is exactly one line long, if the public key was copied successfully.

16.6.6 Testing Key-Based Authentication

At this point, students should have an ed25519 key pair via `ssh-keygen`, and used one of the three methods described in section 16.6.5 to copy their public key file to the `~/.ssh` directory on their bastion host, as well as the SIEM, IPS, and kali VMs. Testing key-based authentication should be as simple as attempting to SSH to each of your lab systems. If students did NOT assign a password to the SSH private key, and everything has been configured properly, the instant students any of the following:

```
ssh bastion_host
ssh siem
ssh ips
ssh kali
```

It should automatically log the student into to the requested system as their configured user automatically with zero password prompts. On the other hand, if students configured their SSH private key with a password in order to set up two-factor authentication, they will need that password every time they attempt to SSH to any of their lab systems. If students run into problems remotely accessing their bastion host or virtual machines, take a look at [section 16.7](#), starting on page 942, for some troubleshooting steps to consider.



```
trobenson@trobinsons-MacBook-Pro .ssh % ssh bastion_host
Enter passphrase for key '/Users/trobenson/.ssh/id_ed25519':
ayy@bastion:~$
```

16-61: At this point, students should test and confirm key-based authentication is working on the bastion host, as well as the SIEM, IPS, and kali virtual machines. Students should attempt connecting to each system over SSH, starting with the bastion host in order to establish SSH forward tunnels.

If students opted to password protect their private key, when they attempt to connect to a host in which key-based authentication has been properly configured, a prompt will appear similar to the one in the illustration above:

```
Enter passphrase for key '[User's home directory]/.ssh/id_ed25519':
```

Upon entering the password assigned to the private key, an SSH session should be established. Be sure to test key-based authentication on all four systems by running the following commands:

```
ssh bastion_host
ssh siem
ssh ips
ssh kali
```

Verify that the SSH session asked for the key passphrase for all four SSH connections.

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
ayy@bastion:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ips
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@siem:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siem
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
ayy@ips:~$
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh kali
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
ayy@Kali:~$
```

16-62: If students did not opt to password protect their SSH private key, then they should be provided with an SSH session instantly, without the need to enter any passwords, provided the public key was copied correctly.

16.7 Troubleshooting SSH connectivity and Key-Based Authentication

If students are here, it means that there are problems either with SSH connectivity in general, or with key-based authentication. This section will cover some troubleshooting steps to follow, and settings to check to help find the root cause of various problems related to SSH connectivity.

As with the vast majority of content in sections [16.5](#) and [16.6](#), a lot of troubleshooting for key-based authentication was covered in chapter 15, [section 15.4](#), pp. 809-812. However, in this section we will be covering how to troubleshoot SSH tunneling as well, since **properly functioning SSH tunnels are a huge requirement to access resources in bare-metal lab environments**, and can be pretty hairy to navigate.

Troubleshooting SSH, and SSH tunnels

- First and foremost, ensure that the bastion host system is powered on and has either a statically assigned IP address, or static DHCP allocation. The IP address of the bastion host should never change.
- The pfSense VM should also be powered on, and the WAN interface IP address should never change, due to the bastion host's static routes that list it as the gateway IP address for the lab network segments (e.g., 172.16.1.0/24 and 172.16.2.0/24).
- In fact, while we're at it, double check to make sure all of the virtual machines are powered on. Yes, it's stupid, and yes, you're going to do it because most network access problems start a layer 1.
- If students are able to establish an SSH session to the bastion host system, log in, and run the `ip route` command. Confirm that there are two lines that look something like this:

```
[management network subnet] via [pfSense WAN interface IP] dev [bastion interface]
[ips network subnet] via [pfSense WAN interface IP] dev [bastion host interface]
```

For example, on my raspberry pi bastion host, `ip route` output looks like this:

```
default via 10.0.0.1 dev eth0 proto dhcp src 10.0.0.7 metric 202
10.0.0.0/24 dev eth0 proto dhcp scope link src 10.0.0.7 metric 202
172.16.1.0/24 via 10.0.0.26 dev eth0
172.16.2.0/24 via 10.0.0.26 dev eth0
```

The **bolded** lines above are the static routes to the lab network. We covered how to configure persistent static routes for Ubuntu using `netplan`, Raspbian using `dhcpcd`, or as an alternative for both Ubuntu and Raspbian, using `/etc/rc.local` in [section 16.3.3](#) (pp. 867-877). Be aware that if your network cable is unplugged from the physical bastion host, or the virtual network interface's link state is messed with in any way for the bastion

host virtual machine, that this could result in static routes disappearing from the routing table. If students decided to persist their static routes through `/etc/rc.local`, these routes may not be regenerated automatically. The quickest way to resolve this problem is to reboot the bastion host.

- The next step is to confirm that the pfSense VM firewall rules on the WAN interface allow the bastion host access to the lab virtual machines. This is why we took so much time creating allow rules on the WAN interface in chapter 14, then modifying those firewall rules to allow access to those network services from the bastion host IP addresses students defined. Review [section 16.3.4](#) (pp. 878-881). Create an IP address alias for your bastion host system(s), then modify the pfSense WAN interface firewall policy to allow access from the bastion host IP address alias. **The pfSense firewall must have firewall rules to allow that network access from the bastion host to any service students want to access remotely.**
- If at this point you've verified that everything else listed here is in working condition, the last thing to do is check the configuration of the SSH tunnels. This was covered for Windows users in sections [16.5.1](#), [16.5.2](#), and [16.5.3](#) (pp. 889-897), and for Linux/macOS users in sections [16.6.1](#), [16.6.2](#), and [16.6.3](#) (pp. 919-931). SSH tunnels to carry the traffic from the student's workstation to the desired host and service must be configured. **If students add additional hosts or services they want to access remotely later, a new firewall rule allow that traffic, and a new SSH tunnel to carry the traffic to the destination will need to be configured.**

Trading SSH Tunnels for A VPN

As students might imagine, adding SSH tunnels to allow remote access to every new system added to the lab environment can become very tedious, as the lab environment continues to grow. If the overhead of managing SSH tunnels becomes too great, it may not be bad idea to learn how to set up the bastion host as a VPN server. While there are many options out there, the most popular free/open-source solutions are OpenVPN, Wireguard, and pivpn (a lightweight OpenVPN solution for raspberry pi systems). Unfortunately, configuring a VPN is outside of the scope of this book, and should be seen as something students progress to, once they get their lab up and running.

- Windows users need to configure a PuTTY session for their bastion host that defines three forward tunnels to the SIEM, IPS, and kali virtual machines, and one dynamic tunnel we'll be covering how to use shortly. When we created this session together in section 16.5.2, we named it `Bastion_Host_Tunnels`. Students should confirm this PuTTY session exists, and ensure its assigned to the bastion host's mRemoteNG connection profile. This is done in the *Config* pane, Under the *Protocol* section, in the field labeled *PuTTY Session*.
 - Windows users may use the string of commands `netstat -ano | findstr LISTEN | findstr 900` to see if the SSH forward and dynamic tunnels are LISTENING for new connections.
 - If ports 9000,9001,9002, and/or 9003 are already in use by another program running on the Windows workstation, this will cause problems with the SSH tunnels. Maybe consider using ports 9030, 9031,9032 and 9033 instead.
 - The mRemoteNG connection profiles for the SIEM, IPS, and kali VMs are unique. In the *Config* pane, the *Hostname/IP* will always be 127.0.0.1, while the *Port* MUST correspond to the forward tunnels configured on the bastion host PuTTY session.
 - For example, a forward tunnel on IP address 127.0.0.1 port 9000 will forward to 172.16.1.3:22 (the SSH service on the SIEM VM).
 - Likewise, for the IPS VM's connection profile, the *Hostname/IP* is 127.0.0.1, and the *Port* should be 9001. For the kali VM, its 127.0.0.1, and *Port* 9002.
- Linux/MacOS users on the other hand, use a `~/.ssh/config` file (or can optionally use aliases containing SSH connection strings) that define how to connect to the bastion host, SIEM, IPS, and kali systems. Check out section 16.6.3, [page 925](#) for a sample SSH config file. Make sure to pay attention to the `LocalForward` and `DynamicForward` directives for the `bastion_host` entry. Ensure that the port numbers and IP addresses used for those directives match the `Port` directive, and the desired lab virtual machine in the `siem`, `ips`, and `kali` entries.
 - If ports 9000, 9001, 9002, and/or 9003 are being used by another application or service on the MacOS or Linux workstation, try using another set of TCP ports (e.g., 9030, 9031, 9032, and 9033) for the `LocalForward`, `DynamicForward`, and corresponding `Port` directives.

- If students are able to establish an SSH session their bastion host, check out the sidebar conversation in section 16.6.2 (pp. 923-924), *Checking your tunnels with netstat and ss*, for directions on how to confirm the forward tunnels are up and listening for connections.

Troubleshooting Key-Based Authentication

- In section 16.5.4 (pp. 898-902), we covered how to generate an SSH keypair: the private key (.ppk file) and the public key, students were instructed to save as `authorized_keys`. In section 16.5.5 (pp. 903-909), students were provided with methods to copy this `authorized_keys` file to the SIEM, IPS and Kali VMs. Windows users need to make sure that the `authorized_keys` file they transfer to their bastion host and lab virtual machines meets the following criteria:
 - The file should be exactly 1 line long. In the margins of *Notepad++*, there is a line count function. Make sure there is only ever one line.
 - The end of line character for the `authorized_keys` file must be configured for Unix line feeds. In *Notepad++*, this is done under *Edit > EOL Conversion > Unix(LF)*.
 - In section 16.5.6 (pp. 910-913), we covered the process of configuring a PuTTY session for the SIEM, IPS, and kali mRemoteNG profiles, and reconfiguring the bastion host's PuTTY session to enable key-based authentication. Ensure that the correct PuTTY sessions are assigned to the correct mRemoteNG connection profiles.
- For Linux and/or MacOS users, it is recommended to use *ssh-copy-id* to automate transferring the SSH public key to the bastion host and lab VMs if possible. It's the safest, fastest, and most effective of the three methods covered in this chapter. This command (and other methods) was covered in section 16.6.5 (pp. 933-939).
- For Windows, Linux or MacOS users, consider the following troubleshooting steps:
 - On the bastion host and lab VMs, run `wc -l ~/.ssh/authorized_keys`. This command should return 1 or 0 as the first character indicating how many lines the file contains. Those are the only two valid values for the file (for now – it is entirely possible for more than one public key to be added to the `authorized_keys` file).
 - On the bastion host and lab VMs, ensure the correct file permissions were set on both the `~/.ssh` directory, as well as `~/.ssh/authorized_keys`, using the commands:
 - `chmod 700 ~/.ssh`
 - `chmod 600 ~/.ssh/authorized_keys`

16.8 Using the Bastion Host as a Web Proxy, using Dynamic Tunnels and FoxyProxy

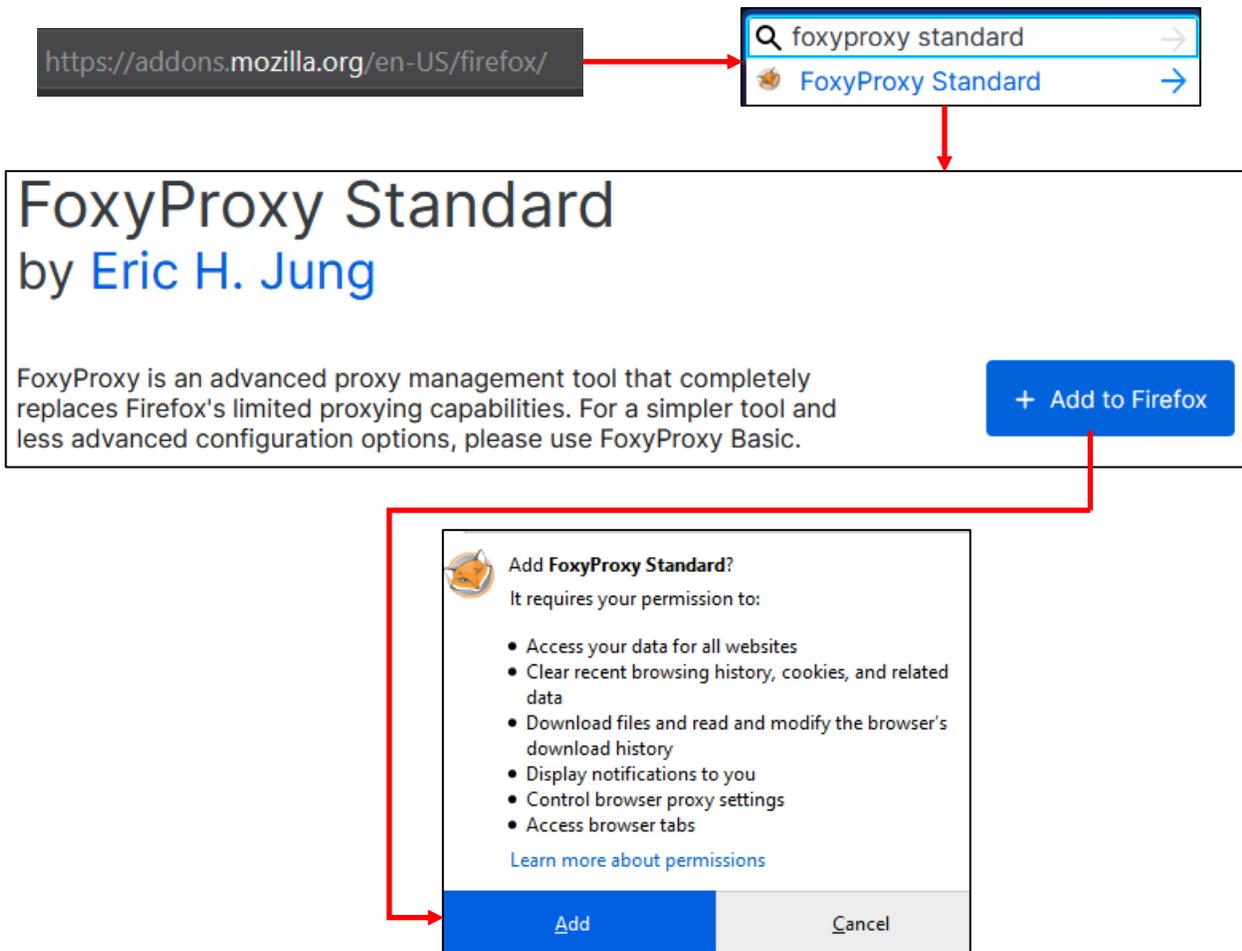
In this section, students will learn how to use the Dynamic SSH tunnel we've configured on the bastion host to treat it like a web proxy. This allows students to access web applications in the lab environment, such as the Splunk web interface we'll be configuring later, and the pfSense webConfigurator.

While most web browsers feature the option to configure a proxy for web services, there is a special add-on for Mozilla Firefox, and most Chrome-based web browsers called *FoxyProxy*. Instead of having to stumble through the settings menu each time students want to use the bastion host as a proxy, then disable the proxy settings by stumbling through the proxy settings again, *FoxyProxy* allows users to configure a network profile and turn that network profile on and off to proxy traffic as needed, without having to dig through the network settings menu so often.

16.8.1 Installation Instructions

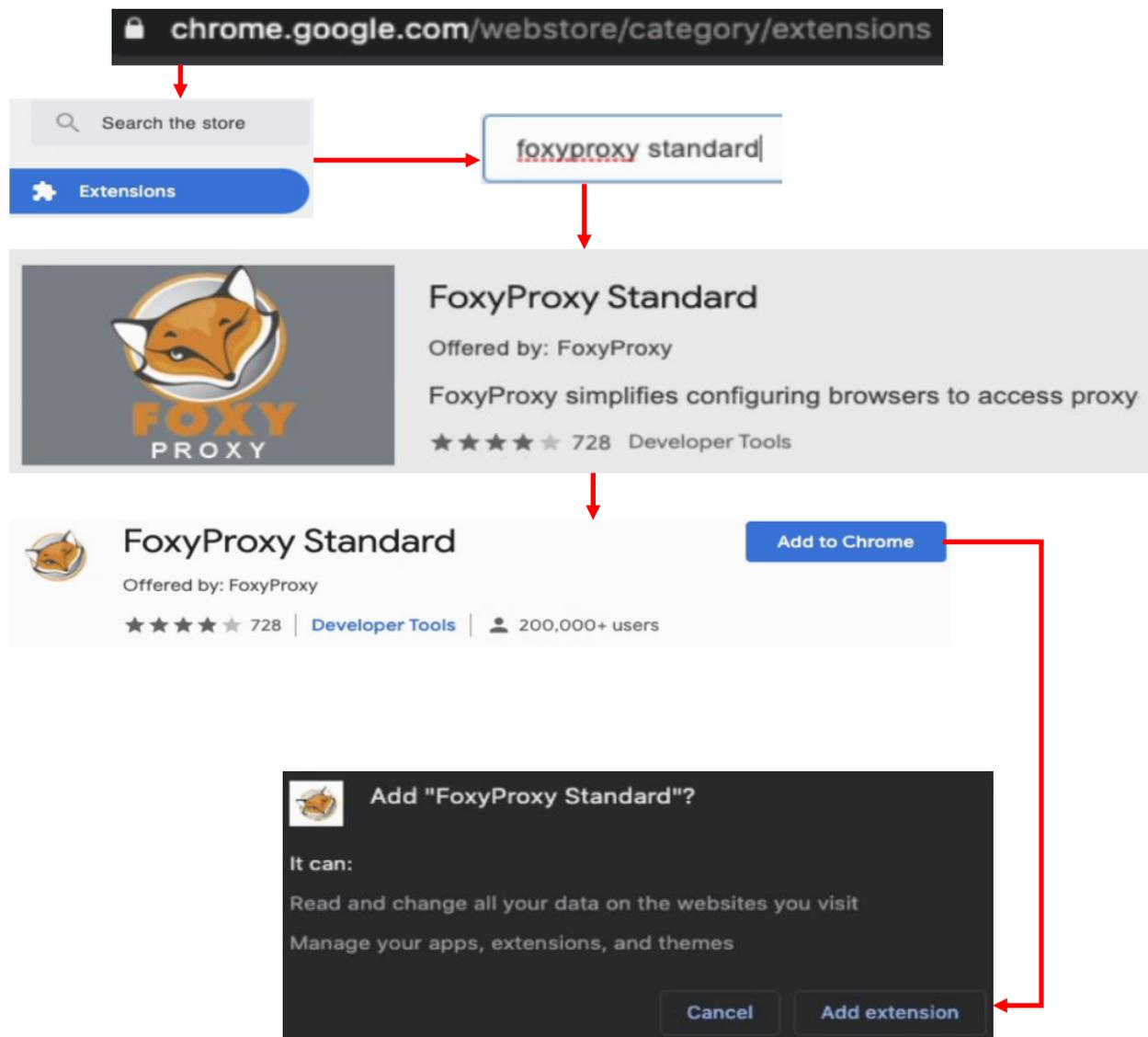
In order to get started, begin by downloading *FoxyProxy Standard* for either Mozilla Firefox (or any of its variant browsers) or Google Chrome (and any of its variant browsers – including Microsoft Edge) using the instructions provided below

Mozilla Firefox: Open Firefox, and navigate to <https://addons.mozilla.org>. Click on the search box marked by a spyglass and enter, "foxyproxy standard". A drop-down menu should appear immediately below the input box. Click on the entry labeled *FoxyProxy Standard* to be taken to the add-on page. Click the button labeled *Add to Firefox*. A pop-up will appear asking if students want to allow the add-on to be installed, along with displaying all of the permissions *FoxyProxy* wants. Click the *Add* button to finish installing the add-on.



16-63: To install *FoxyProxy Standard* on Firefox, visit <https://addons.mozilla.org>. Search for "foxyproxy standard", and click the first link in the drop-down menu below the search bar. On the *FoxyProxy Standard* add-on page, click *Add to Firefox*, then click the *Add* button in the window that pops up in order to confirm installation of the add-on.

Google Chrome: Open Chrome and navigate to <https://chrome.google.com/webstore>. If it is not already highlighted, on the left side of the page, click the option labeled *Extensions* to highlight it. Click on the input box marked with a spyglass to use the site's search function. Type in "foxyproxy standard", then hit enter. The right side of the page should update with search results. Click the entry labeled *FoxyProxy Standard* to continue (it should be the only search result returned). On the *FoxyProxy Standard* extension page, click the *Add to Chrome* button to install FoxyProxy. A pop-up will appear to confirm if students want to install *FoxyProxy Standard*, along with the permissions the extension requires. Click the *Add Extension* button to finish the installation.



16-64: To install *FoxyProxy Standard* on Chrome, visit <https://chrome.google.com/webstore>. Ensure that the *Extensions* option is highlighted on the left side of the page, and in the searchbar marked by a spyglass, enter "foxyproxy standard", then hit enter. In the search results to the right, click on *FoxyProxy Standard* to be taken to the extension page. Click the *Add to Chrome* button, then in the pop-up that appears, click *Add extension* to finish the installation.

16.8.2 Configuration Instructions

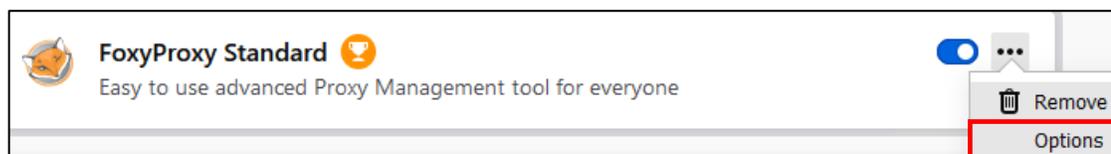
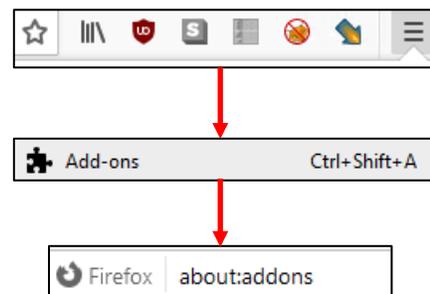
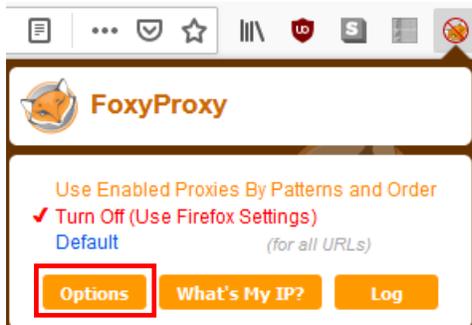
In order for *FoxyProxy* to use our dynamic SSH tunnel to proxy network traffic, we need to access the configuration menu for the add-on. There are a couple of ways to do this, depending on whether students are using Chrome or Firefox, so let's go over them briefly.

Mozilla Firefox: There are two ways to access the *FoxyProxy Standard* options menu.

-*Option 1:* Students may have noticed a small orange icon on their browser's navigation bar that appeared after installing *foxyproxy*. Click on it, and a bunch of options for *FoxyProxy* will appear. Click the button labeled *Options*.

Note: I recommend ensuring that the *FoxyProxy* icon is pinned on the browser's navigation bar. If for some reason it isn't there after installing the add-on, Click the *Open Menu* button in the upper right corner of the browser, click *Customize Toolbar*, find the icon, and drag it to the navigation bar manually.

-*Option 2:* In the far-right corner of the web browser menu, click the *Open Menu* button then click the *Add-ons* option in the drop-down menu that appears. Alternatively, in the browser URL input box, students can enter the special address `about:addons` to be taken to the same configuration menu. In the list of enabled extensions, there should be an entry for *FoxyProxy Standard*. To the right, click the icon with the three dots to bring up a small drop-down menu, then click *Options*.



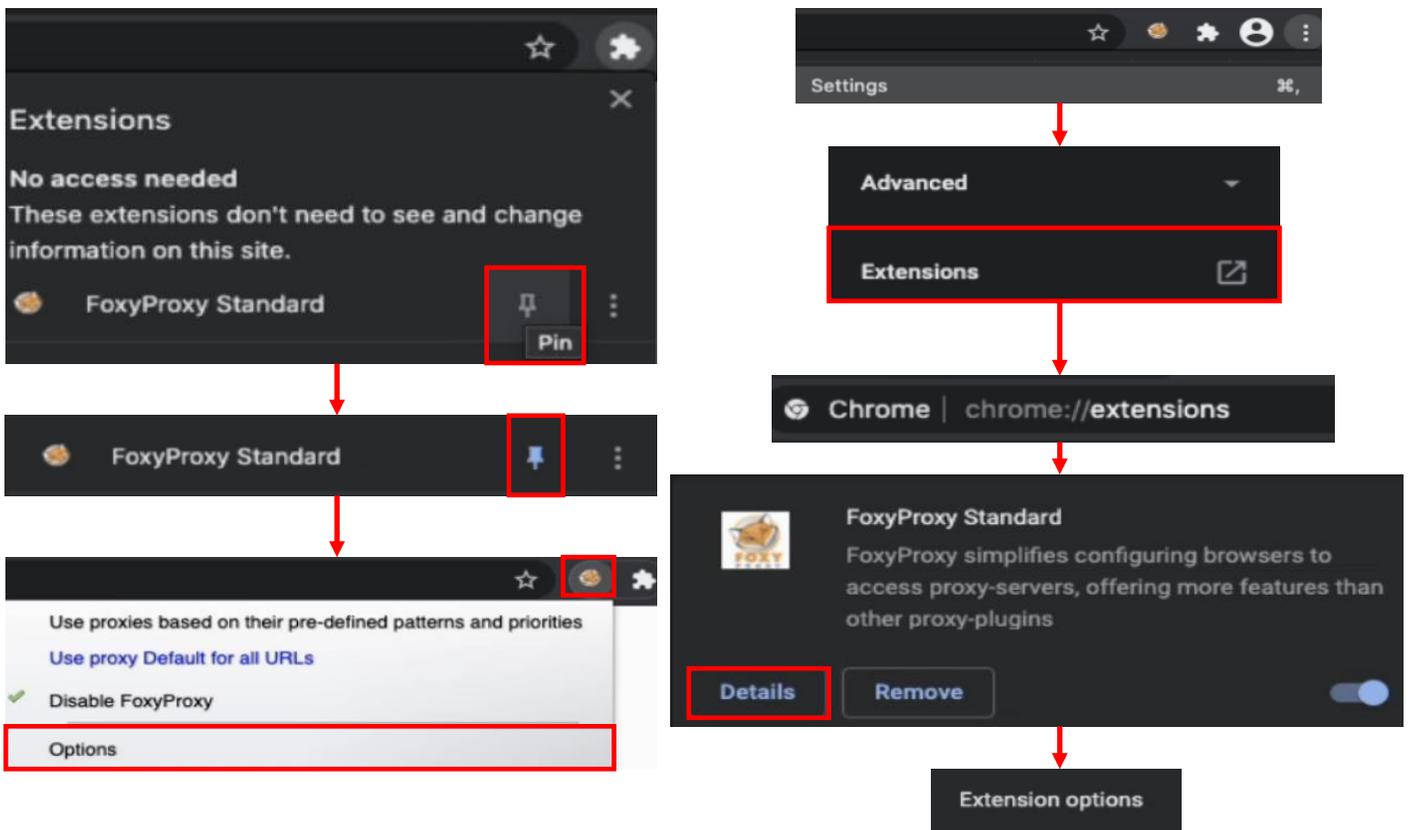
16-65: There are two main ways to reach the options menu for *FoxyProxy Standard*. Find the orange icon in the navigation bar that appears after installing *FoxyProxy*. Click on it, then click on the button labeled *Options*. Or, click on the *Open Menu* button in the browser navigation bar, then click the *Add-ons* option in the drop-down menu that appears. On the `about:addons` page, make sure that *FoxyProxy Standard* is listed under the *Enabled* extensions, then to the right of the *FoxyProxy Standard* entry, click the three dots to bring up a small drop-down menu. Click the *Options* entry.

Google Chrome: Not unlike with Firefox, there are two ways to reach the *FoxyProxy Standard* options menu.

-*Option 1:* In the browser navigation bar, there should be an icon that looks like a puzzle piece click on it, and a list of enabled extensions should appear. I highly recommend clicking the icon that looks like a pushpin to "pin" the *FoxyProxy* icon to the browser's navigation bar. Click on the tiny orange icon that appears on the navigation bar, and a small drop-down menu appears. Click the Options button to proceed.

Note: I highly recommend taking the time now in order to pin the *FoxyProxy* icon to the browser's navigation bar for ease of access.

-*Option 2:* In the far right corner of the web browser, click the three vertical dots to expose a drop-down menu. Click the option labeled *Settings*. In the Chrome settings menu, under *Advanced*, click *Extensions* (optionally, in the navigation bar, students can enter `chrome:extensions` to be taken directly to this menu page). Ensure *FoxyProxy standard* is enabled, then click on the button labeled *Details*. Scroll, down and click *Extension Options*.



16-66: Google Chrome users can click on the puzzle icon, to access the extensions drop-down. Click the push-pin icon to make the *FoxyProxy* icon appear in the navigation bar. Click on the new icon, then click *Options*. Alternatively, click the three vertical dots on in the far-right corner of the browser window, then click *Settings*. In the settings menu, under *Advanced*, click *Extensions*. Locate *FoxyProxy Standard*, ensure it is enabled, then click *Details*. Finally, on the details page, click *Extension options*.

16.8.3 Adding a new proxy, enabling the proxy, and testing connectivity

Once students have made it to the options menu, the instructions for adding a new proxy are more or less the same, regardless of the browser being used, just with some slight visual differences. Click the Add/Add New Proxy button, and use the following configuration settings:

For the Proxy type, ensure that SOCKS5 is selected – For Firefox users, this will be a choice from the *Proxy Type* drop-down menu, while for Google Chrome users, they'll need to check the *SOCKS proxy* checkbox, then click the *v5* radio button.

Make sure that the IP address is set to 127.0.0.1 – In Firefox, enter 127.0.0.1 in to the *Proxy IP address or DNS name* input box, while in Chrome, enter 127.0.0.1 into the *Hostname or IP address* input box.

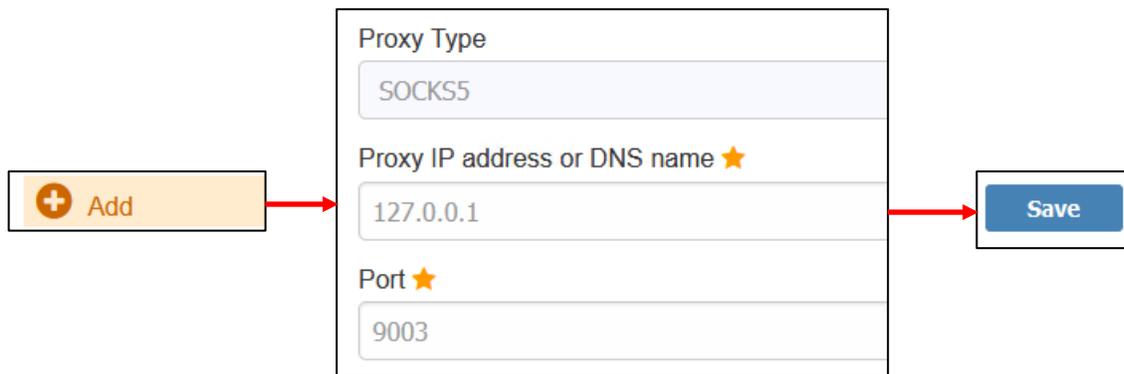
Set the Port input box to the port reserved for the SSH dynamic tunnel – That should be port 9003, unless students have to set their dynamic tunnel to another TCP port number.

Once students have configured the settings above, click the Save button, and exit the options menu for FoxyProxy. In previous section (16.8.2), I provided instructions for students to bind the icon for *FoxyProxy Standard* to the browser's navigation bar. If you haven't already done so, I very highly recommend ensuring the icon is pinned to the navigation bar to ensure quick access to the newly configured proxy profile.

Next, using mRemoteNG (Windows users) or your preferred terminal application (Linux/macOS users), students will need to open an SSH session to their bastion host system in order to activate the Dynamic SSH tunnel needed to proxy their web traffic.

Back in the web browser, click on the FoxyProxy icon, and select our newly created proxy entry. For Firefox, this entry will be named *127.0.0.1: 9003 (for all URLs)*, while Chrome will name the entry *Use proxy 127.0.0.1:9003 for all URLs*.

Next, try to open an HTTPS session to the pfSense WebConfigurator using its WAN IP address. Earlier, students were advised to enter the IP address of their bastion host systems into the Bastion_Hosts IP address alias, then reconfigure the firewall rules on the WAN interface to use this bastion, instead of specific IP addresses to allow access to the lab environment.



Continued to *fig. 16-68*

16-67: The illustrations above describe the process for adding a new proxy to FoxyProxy on Chrome (top) and Firefox (bottom):

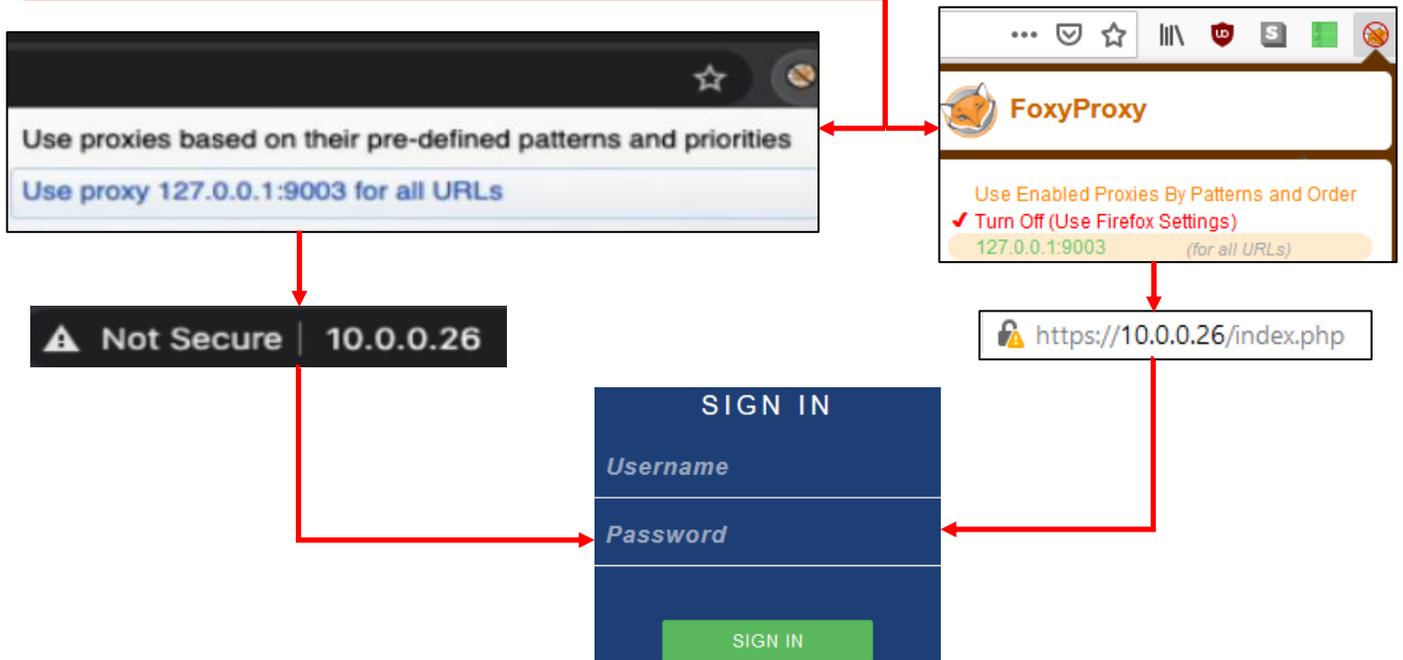
- Click either *Add* or the *Add New Proxy* button
- On the screen that appears, enter the IP address 127.0.0.1 into the *Hostname or IP address* field, or the *Proxy IP address or DNS name* field
- Enter the port students reserved for the SSH dynamic tunnel earlier in this chapter into the *Port* field. Normally, this value should be 9003.

When finished, students can click the *Save* button, and exit the FoxyProxy settings menu.

Continued from *fig. 16-67*

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
```

```
Using username "pi".
Authenticating with public key "ed25519-key-20210107"
Linux raspberrypi 5.10.17+ #1403 Mon Feb 22 11:26:13 GMT 2021 armv6l
```



16-68: Next, students will need to establish an SSH connection to their configured bastion host using either mRemoteNG (Windows) or their preferred terminal application (Linux/MacOS) in order to activate the SSH dynamic tunnel to use as a proxy. Once the SSH session is established, back in the web browser (Firefox or Chrome), click the foxy proxy icon and either click Use proxy 127.0.0.1:9003 for all URLs (Chrome) or 127.0.0.1:9003 (for all URLs) (Firefox).

To begin proxying traffic through the dynamic SSH tunnel. To test the dynamic tunnel, connect to the pfSense interface – [https://\[pfSense VM WAN IP address\]](https://[pfSense VM WAN IP address]). For example, in my lab environment, the IP address of the pfSense WAN interface is 10.0.0.26, so I entered <https://10.0.0.26> in the illustration above.

If the pfSense login screen appears, that means the test was successful, and the traffic is being tunneled correctly.

My Workstation Was Already Allowed to do This...

Earlier in this chapter, we talked about modifying firewall rules on the pfSense VM's WAN interface to allow the bastion host system(s) access to the lab environment through an IP address alias. One of the things I recommended was allowing the IP address of your workstation by adding it to the Bastion_Hosts IP address alias. If you did that, then your workstation should have access to the pfSense webConfigurator regardless of whether or not you configured a dynamic SSH tunnel. You should remove your workstation's IP address from the Bastion_Hosts IP address alias, then test to confirm the SSH dynamic tunnel is working with FoxyProxy.

The reason I recommend this is that we already have the necessary SSH forward tunnels in place in order to SSH to the lab virtual machines, I want students to get familiar with SSH dynamic tunnels, and using them alongside the SSH forward tunnels. The other reason I recommend getting familiar with dynamic SSH tunnels right now is that it will allow easy access to other web applications hosted in the lab environment as well. In a later chapter, students will learn how to set up Splunk on the SIEM VM – including access to a web interface on port 8000/TCP. We already have firewall rules in place that allow access to the SIEM VM on port 8000/TCP, but unless you configured static routes to 172.16.1.0/24, your workstation won't know where to send the packets in order to reach 172.16.1.3:8000. The bastion host already has the necessary static routes, and they should be configured to persist between reboots. Since you already use the bastion host to tunnel SSH traffic to the SIEM, IPS, and Kali VMs, you may as well the dynamic tunnel to proxy your traffic, without having to mess around with static routes on your workstation.

Allow me to demonstrate, using my lab environment as an example. I will demonstrate using mRemoteNG, and Mozilla Firefox on Windows, but this process is fundamentally the same for Linux and MacOS users on Chrome or Firefox.

The WAN IP address of the pfSense VM in my lab is 10.0.0.26. My workstation's IP address is 10.0.0.6. I confirmed that the IP address 10.0.0.6 is not explicitly allowed to access port 443/TCP on the WAN interface, and have also confirmed that the IP address is no longer present in the Bastion_Hosts alias. I try to access the pfSense webConfigurator, and am unable to do so. This means that only the bastion hosts can access the pfSense WebConfigurator.

So using mRemoteNG, I open an SSH session to my bastion host at 10.0.0.7, and establish my SSH tunnels, including the dynamic tunnel. Switching back to my web browser, I click the FoxyProxy icon, and ensure the 127.0.0.1:9003 profile is enabled, and proxying traffic for all URLs. Then in the browser's navigation bar, I input the IP address <https://10.0.0.26> and through the SSH dynamic tunnel I am allowed to access the pfSense webConfigurator once more.

Firewall Aliases IP	
Name	Values
Bastion_Hosts	10.0.0.7, 10.0.0.162
RFC_1918_Addresses	192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8

Rules (Drag to Change Order)						
States	Protocol	Source	Port	Destination	Port	
1 / 6.36 MiB	IPv4 TCP	Bastion_Hosts	*	10.0.0.26	443 (HTTPS)	
0 / 594 KiB	IPv4 TCP	Bastion_Hosts	*	172.16.1.3	22 (SSH)	
0 / 0 B	IPv4 TCP	Bastion_Hosts	*	172.16.1.3	8000	
0 / 411 KiB	IPv4 TCP	Bastion_Hosts	*	172.16.1.4	22 (SSH)	
0 / 537 KiB	IPv4 TCP	Bastion_Hosts	*	172.16.2.2	22 (SSH)	
0 / 1.32 GiB	IPv4+6 *	*	*	*	*	

https://10.0.0.26

The connection has timed out
The server at 10.0.0.26 is taking too long to respond.

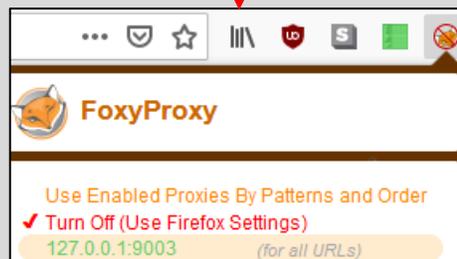
Continued to *fig. 16-70*

16-69: In the interest of making absolutely sure that the SSH dynamic tunnels are working and proxying traffic correctly, make sure that your workstation's IP address is not present in the Bastion_Hosts alias we created on the pfSense webConfigurator earlier. Additionally, make sure that there are no special rules that allow your workstation's IP address direct access to the pfSense webConfigurator.

Confirm you have no access to the pfSense webConfigurator from your workstation by trying to connect to the pfSense WAN interface IP address. For my lab environment the WAN IP address of the pfSense was 10.0.0.26, so in the URL bar, I entered https://10.0.0.26. I received the response, "The Connection has timed out", meaning that the firewall is not allowing the connection.

Continued from *fig. 16-69*

```
Using username "pi".  
Authenticating with public key "ed25519-key-20210107"  
Linux raspberrypi 5.10.17+ #1403 Mon Feb 22 11:26:13 GMT 2021 armv6l
```



 <https://10.0.0.26/index.php>

SIGN IN

Username

Password

SIGN IN

16-70: Next, establish an SSH session to the bastion host system for your lab environment, in order to activate the SSH dynamic tunnel. Next, in your web browser, activate the 127.0.0.1:9003 proxy in FoxyProxy, then attempt to connect to the pfSense WAN IP address, and access the webConfigurator. If you can access the sign in screen, that means the Dynamic SSH tunnel is *definitely* proxying your HTTPS traffic correctly.

The Dynamic Tunnel isn't Working. Now What?

If you're here, it means that for one reason or another, you're unable to access the pfSense webConfigurator using an SSH dynamic tunnel, and FoxyProxy. I'll provide you with a checklist of troubleshooting tasks to perform, borrowing very heavily from the troubleshooting tasks covered in [section 16.7](#):

- Are your virtual machines powered on? Confirm that all of the VMs are powered on, and that none of them are in a bad state (e.g., crashed).
- If the bastion host is a separate, physical host, make sure that it is powered on in a good state as well. Ensure that the network interface is connected to the physical network.
- Confirm that the IP address of the bastion host system is correct.
- Confirm that the IP address of the pfSense WAN interface is correct.
- Log in to the pfSense webConfigurator. If you can't, check out Chapter 14, section 14.4.5.5, the sidebar conversation, [Have you looked in the mailbox?](#) (pp. 721-724) to regain access to the webConfigurator.
 - Under *Firewall > Aliases*, ensure that the bastion host IP address(es) for your lab network are included in the Bastion_Hosts alias.
 - Under *Firewall > Rules*, check the firewall rules for the WAN interface. Ensure that the source for every single allow rule on the interface is set to the Bastion_Hosts alias, and that the Explicit Deny Any firewall rule is at the very bottom of the firewall rule list.
 - These subjects were covered in [section 16.3.4](#) (pp. 878-881)
- Confirm that the SSH connection profile to your bastion host opens three SSH forward tunnels for the SIEM, IPS, and kali Virtual machines, and one dynamic tunnel on ports 9000/TCP through 9003/TCP. That's four SSH tunnels in total.
 - On Windows, make sure that the *PuTTY Session* for the bastion_host mRemoteNG profile is set to Bastion_Host_Tunnels.
 - Check the PuTTY Session configuration under *Tools > Options > Advanced > Launch PuTTY* in order to check the SSH settings. Be sure to load the Bastion_Host_Tunnels section, then check *SSH > Tunnels*.
 - This was covered in [section 16.5.2](#) (pp. 891-895)
 - For Linux/macOS users, this will involve checking the `~/.ssh/config` file and confirming that the profile for the bastion_host is configured correctly.
 - This was covered in [section 16.6.3](#) (pp. 925-931)
- Establish an SSH connection to your lab network's bastion host. After doing so, confirm that the SSH tunnels are in the LISTENING state on your workstation, and waiting for connections:
 - **Windows:** `netstat -an | findstr LISTENING | findstr 900`
 - **MacOS:** `netstat -anp tcp -f inet | grep LISTEN | grep 900`
 - **Linux:** `ss -a1nt4 | grep 900`

- On MacOS and Linux, there should be four entries, while on Windows, there will be eight entries. Confirm ports 9000,9001,9002, and 9003 especially are all present and in the LISTENING status on your workstation.
 - If you were required to use alternate TCP port numbers for your forward/dynamic tunnels, be sure to change the `grep/findstr` commands above to reflect those changes.
- On the ssh session to your bastion host, run the `ip route` command and confirm that there are two entries for the IP address ranges of the lab network segments. For example:
 - `172.16.1.0/24 via 10.0.0.26 dev eth0`
 - `172.16.2.0/24 via 10.0.0.26 dev eth0`
 - Substitute `172.16.1.0/24` and `172.16.2.0/24` if you are using alternate network ranges for the IPS1, IPS2 and Management network segments. Additionally, substitute `10.0.0.26` with the WAN IP address of the pfSense VM as necessary.
- Check the Add-Ons/Extensions of your web browser to ensure that FoxyProxy Standard is installed and enabled. If you are using a web browser that is based on Mozilla Firefox or Google Chrome, there may be additional hoops you have to jump through in order to install/enable Google webstore extensions or Mozilla Add-ons.
- Confirm that the Proxy added to FoxyProxy is configured correctly:
 - The type of proxy should always be SOCKS5
 - The IP address should always be `127.0.0.1`
 - The port number MUST match the TCP port you reserved for the SSH dynamic tunnel. Normally this is port 9003, but if there was already a network service that is using port 9003/TCP, and you used a different TCP port, the *Port* setting needs to be pointing to that TCP port number.
- Make sure that you pinned the FoxyProxy icon to your web browser's navigation bar. Click on the icon, and make sure there is a check mark next to:
 - `127.0.0.1:9003 (for all URLs)` if using Mozilla Firefox
 - *Use proxy 127.0.0.1:9003 for all URLs* if using Google Chrome
 - If you are using an alternate port number, of course substitute that for the options above (e.g., `127.0.0.1:[port number of your dynamic SSH tunnel here]`)

Hopefully this troubleshooting checklist will help you discover why the dynamic SSH tunnel isn't working properly, or at least lead you in the right direction. Remember the OSI network model, and work your way up the layers.

16.9 (Optional Content) Remote Access Enhancements

This section is more or less identical to Chapter 15, section 15.5, adapted for use with bare-metal hypervisors. We'll be talking about a couple of features that, I didn't really feel were "core" to this chapter, but are configuration options that will either allow students to enhance the ease of access to the lab environment, enhance the security of their lab environment, or even strike a balance between accessibility and security. We will cover enabling SSH authentication as the root user, and disabling password authentication for SSH.

Most of the content for this section is identical to the content covered in [section 15.5.1](#) (pp. 813-821), and [15.5.2](#) (pp. 821-830). So in the interest of saving time, and paper, I'll provide you with a bulleted list of things to be aware of, tasks to complete, and/or commands to complete in order to enable SSH access to your lab systems (and/or bastion host) as the root user and/or disable password authentication over SSH.

16.9.1 Enabling SSH Access as the root User

- SSH access as the root user is viewed by many as a systems administration taboo that should never be performed. However, it is also extremely convenient, and combined with key-based authentication (especially if the SSH key is password protected) the downsides can be pretty effectively mitigated.
- Remember that the root user represents absolute authority on most Linux/Unix/BSD systems. Meaning that if anyone else is able to compromise your SSH keys, they can access the server as the root user, and have authority to do pretty much anything. Good thing these are just lab virtual machines that you can snapshot at a moment's notice, but in the real world, key management is a very real problem.
- **If students are interested in enabling SSH access as the root user, key-based authentication should be considered a prerequisite.** Why? On any of the Linux lab VMs and/or the bastion host, run the command:
 - `grep PermitRootLogin /etc/ssh/sshd_config`
 - Pay attention to the line that reads:
 - `#PermitRootLogin prohibit-password`
 - The file `/etc/ssh/sshd_config` contains a variety of configuration parameters that the SSH service reads on startup. In plain English, this line tells the SSH server "Do not allow password authentication as the root user"
- In order to use key-based authentication as the root user, we will copy the `~/.ssh` directory of the standard user account to the root user's home directory, and change file ownership to the root user. Establish an SSH session to the bastion host system, and run the following commands:
 - `sudo su -`
 - `cp -r /home/[username]/.ssh/ ~/`
 - `chown -R root:root ~/.ssh`

- Repeat this process on the SIEM, IPS, and Kali VMs as well.

```
pi@raspberrypi:~ $ grep PermitRootLogin /etc/ssh/sshd_config
#PermitRootLogin prohibit-password
# the setting of "PermitRootLogin without-password".
pi@raspberrypi:~ $ sudo su -
root@raspberrypi:~# cp -r /home/pi/.ssh/ ~/
root@raspberrypi:~# chown -R root:root ~/.ssh
root@raspberrypi:~# exit
logout
pi@raspberrypi:~ $ exit
```

16-71: In order to enable SSH login as the root user, students need to confirm that its even allowed by SSHD in the first place. First, check the PermitRootLogin directive by using the grep command to search for that string in /etc/ssh/sshd_config. In the illustration above, it is confirmed that the default setting is prohibit-password, meaning that in order to log in as root, students will need to establish key-based authentication for the root user.

Fortunately, students should have already configured key-based auth for the standard user account already, so its just a matter of copying the standard users .ssh directory to the root user's home directory, then changing the file ownership so that root owns the .ssh directory and all of the files contained within – specifically, the authorized_keys file. After completing this process on the bastion host, students may repeat these steps on the SIEM, IPS, and Kali virtual machines, as well.

16.9.1.1 Testing root SSH for Linux/MacOS Users

- After running the commands specified in section 16.9.1 on the bastion host, SIEM, IPS, and kali systems, open new terminal sessions and run the following commands:
 - ssh root@bastion_host
 - ssh root@siem
 - ssh root@ips
 - ssh root@kali

```
trobinson@trobinsons-MacBook-Pro ~ % ssh root@bastion_host
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
root@bastion:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh root@siem
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
root@siem:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh root@ips
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
root@ips:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh root@kali
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
root@Kali:~#
```

16-72: Run the following commands:

```
ssh root@bastion_host
ssh root@siem
ssh root@ips
ssh root@kali
```

to test whether or not key-based authentication as the root user was configured properly. Depending on whether students opted to password protect their private keys, they will either be prompted for the password to decrypt the private key, or instantly provided with an SSH session as the root user, if everything is working properly.

- If students want to be automatically logged in as the root user, it's possible to create multiple SSH config Host statements for the same IP address, just logging in as different users..
- For example, a user could create a Host siem profile to ssh to the siem VM as the ayy user, and a Host siemroot profile to SSH to the SIEM VM as the root user.
- Below is a sample ~/.ssh/config file students can copy that contains profiles to log in as the root user on the bastion host, SIEM, IPS, and Kali virtual machines:

SSH config file with root user connection profiles

```
Host bastion_host
    Hostname 10.0.0.162
    User ayy
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003
```

```
Host bastion_host_root
    Hostname 10.0.0.162
    User root
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003
```

```
Host siem
    Hostname 127.0.0.1
    Port 9000
    User ayy
```

```
Host siemroot
    Hostname 127.0.0.1
    Port 9000
    User root
```

```
Host ips
    Hostname 127.0.0.1
    Port 9001
    User ayy
```

```
Host ipsroot
    Hostname 127.0.0.1
    Port 9001
    User root
```

```
Host kali
    Hostname 127.0.0.1
    Port 9002
    User ayy
```

```
Host kaliroot
    Hostname 127.0.0.1
    Port 9002
    User root
```

- As usual substitute `User` with the username students configured for the bastion host, SIEM, IPS, and Kali systems.
- This configuration file also assumes the bastion host's IP address is 10.0.0.162. Chances are, this is not the case for students' lab environments. To fix this, change the `Hostname` field for the `bastion_host` and `bastion_host_root` entries as necessary.
- If students are using a different set of subnets for their lab environment other than 172.16.1.0/24 and 172.16.2.0/24, then the `LocalForward` entries for both the `bastion_host`, and `bastion_host_root` entries will need to change to reflect the IP addresses for the SIEM, IPS and Kali virtual machines.
- Additionally, if students are using different ports for the forward tunnels other than ports 9000, 9001, 9002, and 9003, be sure to change the `LocalForward` and `DynamicForward` entries for the `bastion_host` and `bastion_host_root` entries, as well as the `Port` field for the `siem`, `siemroot`, `ips`, `ipsroot`, `kali`, and `kaliroot` entries to reflect the changed port numbers.
- A copy of this file is available at:
 - <https://gist.github.com/da667/802c5bb2be78a52f9a57bde03483c1c2>

```
trobinson@trobinsons-MacBook-Pro ~ % cat ~/.ssh/config
Host bastion_host
    Hostname 10.0.0.162
    User ayy
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003

Host bastion_host_root
    Hostname 10.0.0.162
    User root
    LocalForward 9000 172.16.1.3:22
    LocalForward 9001 172.16.1.4:22
    LocalForward 9002 172.16.2.2:22
    DynamicForward 9003

Host siem
    Hostname 127.0.0.1
    Port 9000
    User ayy

Host siemroot
    Hostname 127.0.0.1
    Port 9000
    User root

Host ips
    Hostname 127.0.0.1
    Port 9001
    User ayy

Host ipsroot
    Hostname 127.0.0.1
    Port 9001
    User root

Host kali
    Hostname 127.0.0.1
    Port 9002
    User ayy

Host kaliroot
    Hostname 127.0.0.1
    Port 9002
    User root
```

Continued to *fig. 16-74*

16-73: If students wish to further enhance their laziness, they can create specific Host profiles in their SSH config file that will allow them to connect to the bastion host, SIEM, IPS and/or Kali systems as the root user, so long as key-based authentication has been properly enabled on each of those virtual machines.

Continued from *fig. 16-73*

```
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host_root
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-66-generic x86_64)
root@bastion:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh siemroot
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-65-generic x86_64)
root@siem:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh ipsroot
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-65-generic x86_64)
root@ips:~#
```

```
trobinson@trobinsons-MacBook-Pro ~ % ssh kaliroot
Linux Kali 5.8.0-kali1-amd64 #1 SMP Debian 5.8.7-1kali1 (2020-09-14) x86_64
root@Kali:~#
```

16-74: The modified `~/ .ssh/config` file enables students to run:

```
ssh bastion_host_root
ssh siemroot
ssh ipsroot
ssh kaliroot
```

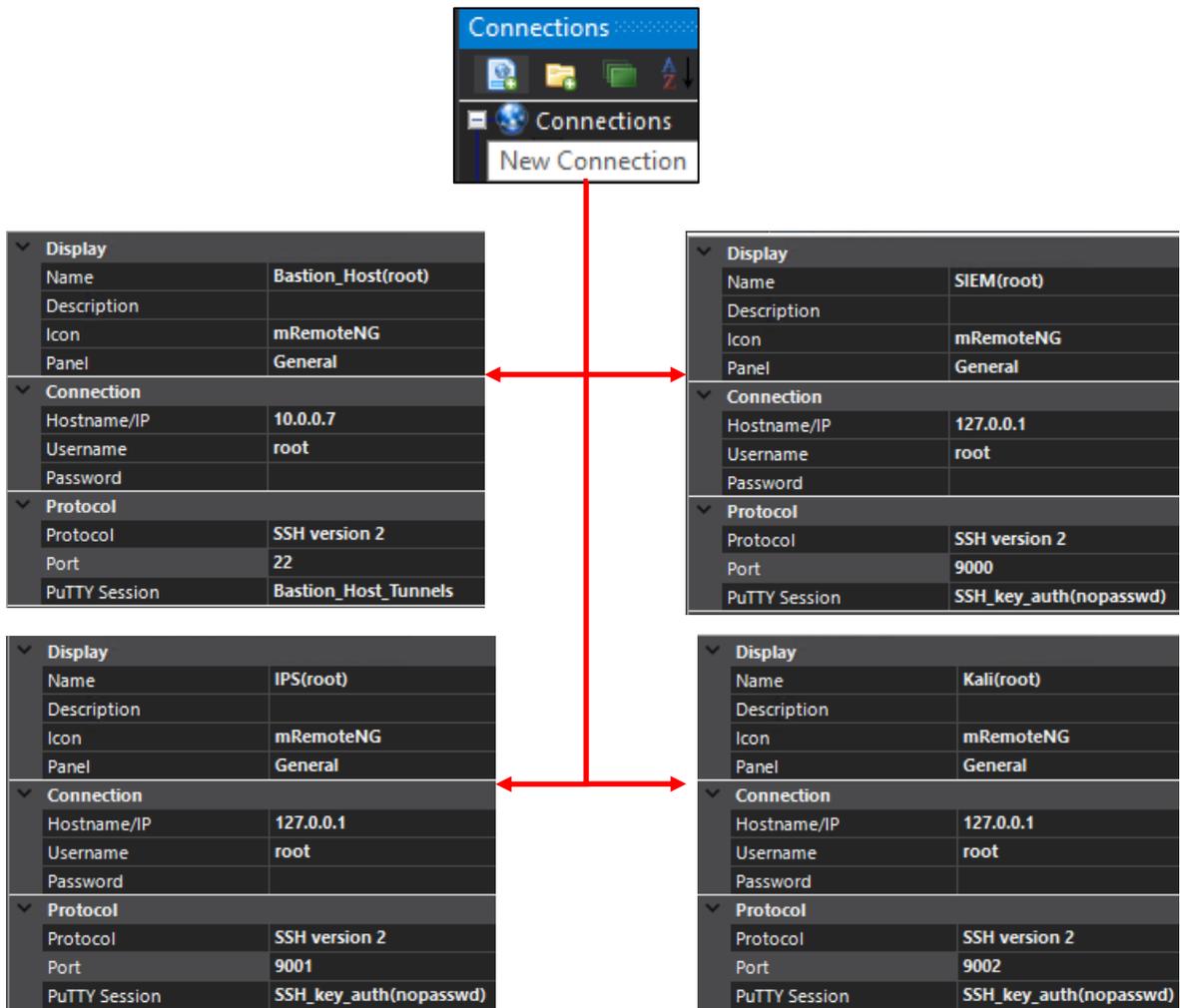
And log in as the root user via key-based auth automatically. Recall that to connect to the SIEM, IPS, and kali VMs, that the SSH session to the bastion host *must* remain active. Additionally, the SSH config file also enables shortcuts for the `scp` command as well, meaning students can copy files to and from their lab virtual machine as the root user if necessary.

16.9.1.2 Testing root SSH for Windows Hypervisor Hosts

- In order to test connectivity as the root user from a Windows workstation, mRemoteNG will require new connection profiles. Create 4 new mRemoteNG connection profiles with the following information:

Name	Bastion_Host(root)	SIEM(root)	IPS(root)	Kali(root)
Hostname/IP	10.0.0.7	127.0.0.1	127.0.0.1	127.0.0.1
Username	root	root	root	root
Protocol	SSH Version 2	SSH Version 2	SSH Version 2	SSH Version 2
Port	22	9000	9001	9002
PuTTY Session	Bastion_Host_Tunnels	SSH_key_auth (nopasswd)	SSH_key_auth (nopasswd)	SSH_key_auth (nopasswd)

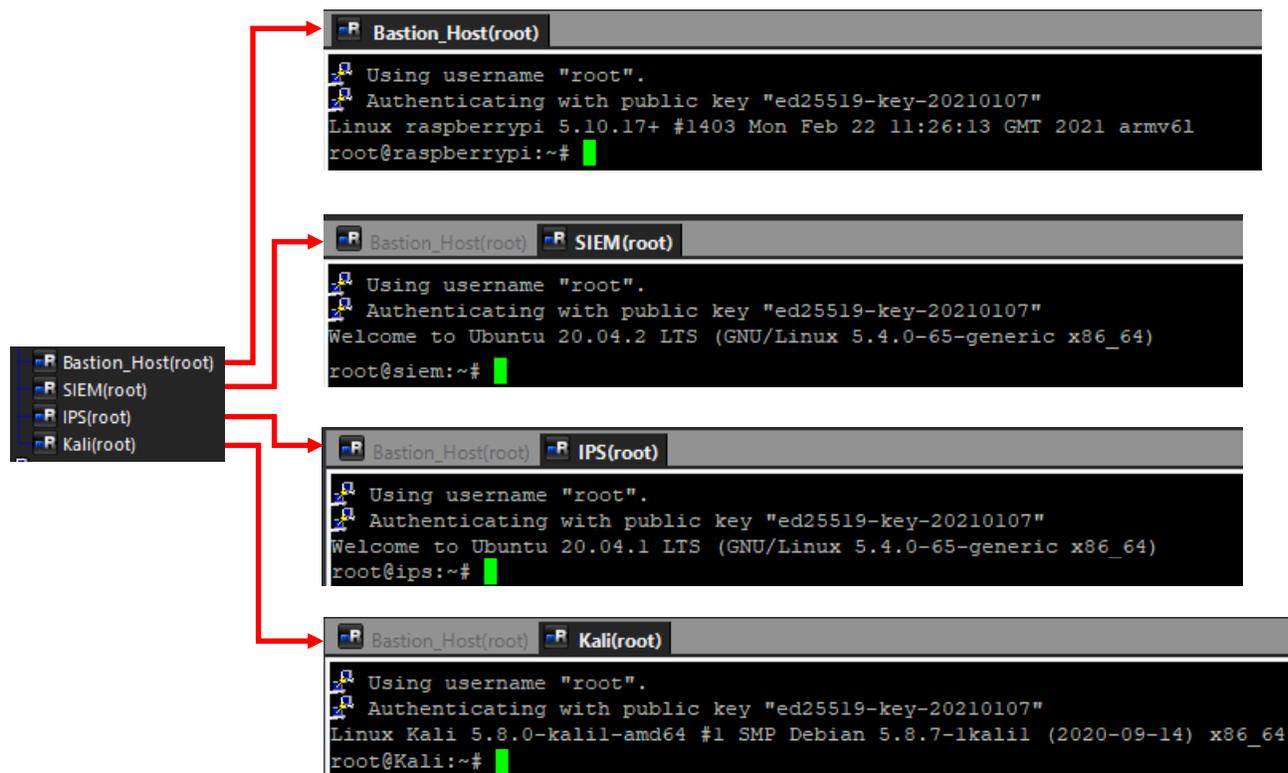
- Students should modify the Hostname/IP field of the Bastion_Host(root) profile to reflect the IP address assigned to the bastion host on their lab network
- Be aware that the name of the PuTTY Session students created may differ, especially if students elected to password protect the SSH private key file.
- Confirm that the Bastion_Host_Tunnels PuTTY Session has key-based auth enabled. We covered this topic in [section 16.5.6](#) (pp. 910-913), otherwise attempting to use this connection profile will fail.
- Make sure that the port numbers assigned to the SIEM(root), IPS(root), and Kali(root) mRemoteNG profiles matches the ports reserved for the SSH forward tunnels in the Bastion_Host_Tunnels PuTTY Session. This was covered in [section 16.5.2](#) (pp. 891-895)



Continued to *fig. 16-76*

16-75: Windows students on the other hand will need to create new mRemoteNG connection profiles to test whether or not root access over SSH is working. Create the connection profiles, using the table on the previous page as a guide for the configuration settings to use.

Continued from *fig. 16-75*



16-76: With the mRemoteNG connection profiles created, double-click on each of them to test them out, making sure that the Bastion_Host(root) SSH connection remains active for the SSH forward tunnels to function properly.

16.9.1.3 Remember, This isn't Strictly Necessary

In future chapters, certain software will require root access to be installed or run properly. However, if you are not comfortable with enabling SSH access for the root user that it is not necessarily. Remember that `sudo su -` is a perfectly viable alternative to elevate to the root account as needed. However, **regardless of whether or not students intend to configure SSH access as the root user, I would highly advise enabling the security enhancements in section 16.9.2 to disable password authentication over SSH entirely for all users.**

16.9.2 Disabling password authentication over SSH

- Disabling password authentication over SSH involves making a handful of configuration changes to the `/etc/ssh/sshd_config` file – specifically the `ChallengeResponseAuthentication`, `AuthenticationMethods` and `PasswordAuthentication` directives.
- In this section, students will:
 - Make a backup of the `/etc/ssh/sshd_config` file
 - Check the current configuration settings for `ChallengeResponseAuthentication`, `AuthenticationMethods`, and `PasswordAuthentication` using the `egrep` command
 - Make changes to the `/etc/ssh/sshd_config` file to support our goal of disabling password authentication, and restart the `ssh` service to apply the configuration file changes.
- Students should be aware that if they enable these security enhancements then somehow misplace or lose their SSH private keys (and/or the password to any password-protected private keys) that they will effectively lock themselves out of SSH access to their SIEM, IPS, and Kali virtual machines.
 - This is why I will guide students through backing up the `sshd_config` file, and/or restoring the backup if necessary.
 - **Alternatively, if there are no backups of the `sshd_config` file available, or access via the virtual console is not possible, taking a virtual machine snapshot prior to making these changes is also recommended.**

16.9.2.1 Backing Up (and Restoring) the `/etc/ssh/sshd_config` file

- To make a backup of the `/etc/ssh/sshd_config` file, performing the following tasks:
 - Open SSH sessions to the bastion host, SIEM, IPS, and Kali systems. Preferably as the root user, but if SSH access as the root user has not been enabled, students can use `sudo su -` to become the root user.
 - Run the following commands as the root user:
 - `cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old`
 - To re-iterate, perform this task on the bastion host, as well as the SIEM, IPS, and Kali virtual machines.

- If for some reason students need to restore the `sshd_config` back to its default state, run the following commands as the root user:
 - `cp /etc/ssh/sshd_config.old /etc/ssh/sshd_config`
 - `systemctl restart sshd.service`

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.old
ls -al /etc/ssh/sshd*
-rw-r--r-- 1 root root 3316 Aug 19 00:08 /etc/ssh/sshd_config
-rw-r--r-- 1 root root 3316 Feb 12 00:45 /etc/ssh/sshd_config.old
```

```
cp /etc/ssh/sshd_config.old /etc/ssh/sshd_config
systemctl restart ssh.service
```

16-77: Since students are about to make significant changes to `/etc/ssh/sshd_config`, it is recommended to make a backup of this file on the bastion host, SIEM, IPS, and Kali systems before modifying it. The first screen capture above is a single command that, when ran as the root user, allows students back up the `sshd_config` to `sshd_config.old` prior to making any modifications.

The second image illustrates the process of restoring the default `ssh_config` from the `sshd_config.old` file, then restarting the SSH service so that SSH service reads and applies the "new" configuration. This process can be done in the event that users lock themselves out of SSH access in the future, and want to restore access without resorting to reverting an old VM snapshot.

16.9.2.2 Modifying the `PasswordAuthentication`, `ChallengeResponseAuthentication`, and `AuthenticationMethods` directives

- Open SSH sessions to the bastion host, SIEM, IPS, and Kali systems. Preferably as the root user, but if SSH access as the root user has not been enabled, students can use `sudo su -` to become the root user. Then, run the following command:
 - `egrep "ChallengeResponseAuthentication|AuthenticationMethods|PasswordAuthentication" /etc/ssh/sshd_config`
 - Any virtual machine based on Ubuntu Server 20.04 (SIEM, IPS and/or bastion host running Ubuntu) will produce the output in *fig. 16-78* below:

```
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
PasswordAuthentication yes
```

16-78: These are the results from the `egrep` command that students should see for any physical or virtual system running Ubuntu Server 20.04. There should be no results for `AuthenticationMethods`.

- From this, we can determine that ChallengeResponseAuthentication is already set to no, PasswordAuthentication is set to yes, and AuthenticationMethods has not yet been defined. Let's change that by running the following commands as the root user on the bastion host system (if its running Ubuntu Server 20.04), the SIEM VM, and the IPS VM:
 - sed -i 's#^PasswordAuthentication yes#PasswordAuthentication no#' /etc/ssh/sshd_config
 - echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config
 - echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
 - egrep "ChallengeResponseAuthentication|PasswordAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
 - systemctl restart ssh.service

```
~# sed -i 's#^PasswordAuthentication yes#PasswordAuthentication no#' /etc/ssh/sshd_config
~# echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config
~# echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
PasswordAuthentication no
#Adding AuthenticationMethods publickey to force key-based auth over SSH only.
AuthenticationMethods publickey
~# systemctl restart ssh.service
```

16-79: As the root user, use the sed command to change PasswordAuthentication from yes to no, and the echo command with output redirection to add the AuthenticationMethods publickey parameter. Next, egrep is used to confirm that the sed and echo commands modified the configuration file correctly. Finally, students restart the SSH service so that the service will read and implement the changes in the sshd_config file. Remember, these specific sed commands should only be ran on bastion host (if its running Ubuntu server), the SIEM VM, and the IPS VM.

- However, the output from the egrep command changes slightly for the Kali VM, as well as bastion host systems running Raspbian (raspberry pi):

```
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
#PasswordAuthentication yes
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
```

16-80: Notice that the PasswordAuthentication directive includes a "#" symbol directly in front of it. This sshd_config is functionally identical to the sshd_config on our Ubuntu systems (bastion host, SIEM and IPS). However, the sed command used to edit that line for Raspbian (raspberry pi bastion hosts) and/or the Kali VM will need to be modified slightly. Once again, ChallengeResponseAuthentication should default to no, and AuthenticationMethods should not be set.

- Run the following commands on the Kali VM, and/or any bastion host systems running Raspbian as the root user:
 - `sed -i 's/^#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config`
 - `echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config`
 - `echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config`
 - `egrep "ChallengeResponseAuthentication|PasswordAuthentication|AuthenticationMethods" /etc/ssh/sshd_config`
 - `systemctl restart ssh.service`

```

~# sed -i 's/^#PasswordAuthentication yes/PasswordAuthentication no/' /etc/ssh/sshd_config
~# echo "#Adding AuthenticationMethods publickey to force key-based auth over SSH only." >> /etc/ssh/sshd_config
~# echo "AuthenticationMethods publickey" >> /etc/ssh/sshd_config
~# egrep "PasswordAuthentication|ChallengeResponseAuthentication|AuthenticationMethods" /etc/ssh/sshd_config
PasswordAuthentication no
ChallengeResponseAuthentication no
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
#Adding AuthenticationMethods publickey to force key-based auth over SSH only.
AuthenticationMethods publickey
~# systemctl restart ssh.service

```

16-81: The slight change in the Kali Linux and Raspbian `sshd_config` file means we have to use a slightly different `sed` command to change the `PasswordAuthentication` directive to `no` on the Kali VM. Other than that, the remaining commands are identical to the ones ran on the SIEM and IPS virtual machines and/or Ubuntu server bastion hosts.

You sed it, brother

Please note that students do not have to use the `egrep` and/or `sed` commands to view or modify the necessary lines in the `sshd_config` file on the bastion host, SIEM, IPS, or Kali systems. If you're comfortable reading files using other command line utilities such as `less`, `more`, `cat`, etc, you're more than welcome to use any other utilities you are more comfortable with. Likewise, if you'd rather modify the necessary lines in the `sshd_config` file on the lab virtual machines using other text editors, students are welcome to use `vi`, `nano`, `ed`, `emacs` or any other text editor they are comfortable with to modify the configuration files as necessary.

If you're open to suggestion, it's worth learning how to edit configuration files using `vi`, mainly because every Linux/Unix-based system has it by default. Not to mention, these configuration directives may change again in the future. What better place to learn a text editor than in your lab environment?

No matter what method or utilities you use to modify the the `sshd_config` file, make absolutely sure that you made a backup of the original file before you begin. That way if there is a problem, there is a known good backup to restore from.

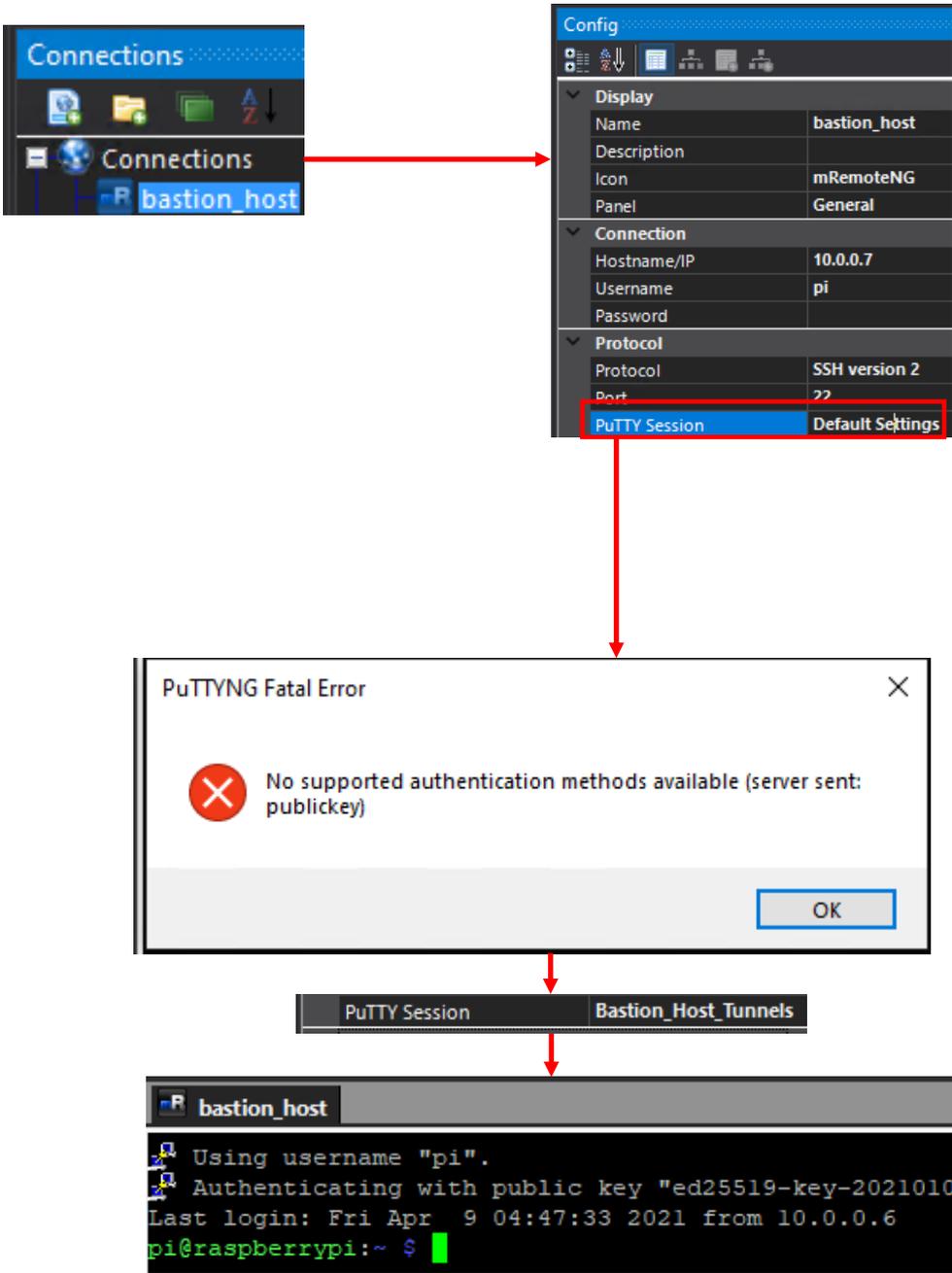
16.9.2.3 Verifying Password Authentication over SSH is disabled

Windows Users:

- Open mRemoteNG and select the `bastion_host` connection profile in the *Connections* pane.
- In the *Config* pane, change *PuTTY Session* from `Bastion_Host_Tunnels` back to `Default Settings`.
- Double click on the `bastion_host` connection profile. Notice that you are no longer prompted to enter a password for the bastion host user, but instead are instantly disconnected.
- Change the *PuTTY Session* setting back to `Bastion_Host_Tunnels`

Note: We covered this in the sidebar conversation in Section 16.5.7. However, it is important enough that this bears repeating: **If you decided to disable password authentication over SSH on your lab virtual machines, WinSCP will no longer work without configuring key-based authentication.** If you followed the instructions for disabling password authentication via SSH to the lab virtual machines, make absolutely sure you read the sidebar conversation in section 16.5.7, *Bonus Lesson: Key-Based Authentication with WinSCP* (pp. 917-918), and **enable key-based auth for your WinSCP site profiles, if you need to transfer files to your lab virtual machines later.**

Students should be able to once more successfully establish an SSH session to the bastion host. Repeat this process on the SIEM, IPS, and Kali virtual machines to ensure password authentication over SSH is disabled on those systems as well.



16-82: The method for testing if password authentication over SSH is disabled for Windows clients is very simple. Modify the `bastion_host` connection profile's `PuTTY Session` setting back to `Default Settings`, then attempt to connect to the `bastion_host` over SSH. If students get an error notification stating there are no supported authentication methods available, everything is working as intended. Students can then change the `PuTTY Session` setting back to the session created earlier to enable key-based authentication, and verify SSH access is working properly once more. Repeat this process for the SIEM, IPS, and Kali virtual machines.

Linux/MacOS Users:

- Run the following commands
 - `mv ~/.ssh/id_ed25519 ~/.ssh/id_ed25519.old`
 - `ssh bastion_host`
- If everything is working properly, students will get a single line error message:

```
[username]@[bastion host IP address]: Permission denied (publickey).
```

If this error message appears then that means the configuration changes were successful! Next, run the following commands:

```
mv ~/.ssh/id_ed25519.old ~/.ssh/id_ed25519
ssh bastion_host
```

Students should be able to once more successfully establish an SSH session to the bastion host. Repeat this process on the SIEM, IPS, and Kali virtual machines to ensure password authentication over SSH is disabled on those systems as well.

Note: I'm going to give you Linux/MacOS users mostly the same conversation I gave the Windows users above: the `scp` command, or an SCP/SFTP client is going to be the best and most secure method of copying files to or from your lab environment. **Ensure whatever SCP/SFTP client you're using has a way for you to specify SSH private keys to use for authentication, if you choose to disable password authentication over SSH.** Fortunately, the `scp` command knows to do key-based authentication by default, and knows how and where to find the SSH private keys (`~/.ssh`) without any further setup.

```
trobinson@trobinsons-MacBook-Pro ~ % mv ~/.ssh/id_ed25519 ~/.ssh/id_ed25519.old
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@10.0.0.162: Permission denied (publickey).
trobinson@trobinsons-MacBook-Pro ~ % mv ~/.ssh/id_ed25519.old ~/.ssh/id_ed25519
trobinson@trobinsons-MacBook-Pro ~ % ssh bastion_host
ayy@bastion:~$
```

16-83: In order to test whether or not password authentication over SSH is disabled on the bastion host, use the `mv` command to rename the `id_ed25519` file (the private key) to literally anything else. In this case, `id_ed25519.old`. Afterwards, run `ssh bastion_host`. Students should immediately get an error message `Permission denied (publickey)`. Use the `mv` command to rename the `id_ed25519.old` back to `id_ed25519`. Once more, run the command `ssh bastion_host` and verify the ability to successfully connect to the SIEM VM over SSH, using key-based authentication.

Laziness, or Security?

There are a lot of die-hard cybersecurity professionals out there who will tell you that every single SSH key you generate and use should be password protected. Well, I don't necessarily agree with that. I believe in presenting you with choices, and demonstrating the trade-offs.

This is a lab environment that is meant to be easy to stand up and modify at a moment's notice. If it's annoying, or otherwise difficult to access your VMs, you're not going to want to use this environment, right? If you set up key-based authentication without password protecting your private key, it's extremely fast and convenient to access your virtual machines remotely. Fast and easy remote access to your systems so you can focus on whatever projects you're working on!

But if that's the case, why am bothering showing you how to password protect your SSH key and set two-factor authentication at all? Because, in the real world, it's considered the right way™ to do key-based authentication. I'm showing you how to do things the right way™ so that when you get out there in a real world, you're aware of how to set it up quickly and safely. What better place to make sure you got the basics down than in your lab environment?

Bear in mind, we didn't cover some of the more arcane and maddening authentication methods that SSH can support (for instance, TOTP challenge/response, push authentication, and Yubikey/FIDO authentication), but I'll leave those as things you can research and look in to on your own.

16.10 Chapter Review

Congratulations on making it to the end of another chapter! The purpose of this chapter was to set bare-metal hypervisor users (e.g., VMware ESXi) up for success for the remaining lab configuration chapters. By the end of this chapter, students will have gained a better understanding of:

- Bastion hosts, and their role for controlling access to the bare-metal lab environment
- How to set up a bastion host virtual machine, or raspberry pi computer (or any other physical computer, for that matter).
- How to configure static routing for the bastion host.
- How to incorporate the bastion host as an access control mechanism for the bare-metal lab environment, through TCP Forwarding (aka "SSH Tunneling") connection requests to the lab environment through the bastion host.
- How to configure the pfSense firewall to restrict access to the lab environment to where access through the bastion host is mandatory.
- The difference between SSH Forward, Reverse, and Dynamic tunnels.
- Utilizing the Secure Shell (SSH) protocol to enable secure and convenient remote access to their lab virtual machines. Students on Windows got familiar with mRemoteNG and setting up connection profiles, while Linux/macOS students learned more about the `ssh` command and setting up the SSH `config` and `Host` profiles.
- Students also learned how they were expected to "tunnel" their traffic through their bastion host system in order to establish SSH access to their lab virtual machines.
- Key-based authentication for SSH. Students learned how to generate their own public/private key pairs, and how key-based authentication can be used to either enhance the security of their SSH connectivity through two-factor authentication (*something they have* – their SSH private key files, and *something they know* – the password required to decrypt those private key files) or to further enhance the convenience of remote access via SSH by no longer requiring any passwords at all.
- Students learned the benefits and risks that come with their decisions regarding private key security. Additionally, students also learned how to configure key-based authentication on the bastion host, SIEM, IPS and Kali systems – including how transfer the SSH public key to remote systems, as well as the particular location, and file permissions required for key-based authentication to function properly.
- How to configure their web browser to utilize an SSH dynamic tunnel on the bastion host as a web proxy, using the Chrome/Firefox add-on FoxyProxy.
- How to enable SSH access as the `root` user. Students learned about the advantages and risks that come with allowing the `root` user to log in remotely using the SSH protocol as well as ways to limit that risk.
- How to disable password authentication over SSH entirely for the bastion host SIEM, IPS and Kali systems, to better enhance the security of their lab environment.

Here is a list of tasks left for students to finish their lab environment:

- Students still need to install either the Snort3 or Suricata IDS/IPS software to enable network access to the Metasploitable 2 VM, and IPS 2 network segment. This process is covered in chapter 17, *Network Intrusion Detection*, starting on p. 980.
- The SIEM VM needs to have Splunk installed and configured, and the IPS VM will need to have log forwarding enabled. This is covered in chapter 18, *Setting up Splunk*, starting on p. 996.
- Are you looking for some ideas on how you can customize your lab environment? Check out chapter 19, *End of the Beginning*, starting on p. 1037 for some recommendations.
- I created a small bonus chapter that contains content that may be useful to help harden your lab environment, and automate keeping most of your VMs up to date. Go check out chapter 20, *Extra Credit*, starting on p. 1055.

Chapter 17 Patch Notes

-Probably the biggest change in this chapter from the first book is coverage for Snort 3.

-I promised myself I wouldn't cover snort 3 until it hit general availability. Lo and behold after being in development for over a decade, they finally released it in January 2021.

-Not a whole has changed with Autosuricata, which is a testament to how down pat the installation process is

-Added a troubleshooting section to this chapter to help students work their way through difficulties with either the installation shell script, or the IDS/IPS software not functioning correctly after install.

-I wanted to make sure that I gave special recognition to Noah Dietrich His Snort 3 installation guide for Ubuntu 18.04 and 20.04. His work was the foundation for Autosnort3. What's more is that his installation guide has a very succinct guide for installing Splunk that I'll be making heavy use of in Chapter 18. I asked for Noah's permission to utilize the documentation he has written as a part of this book and he consented.

-For anyone wanting to take a look at Noah's work, the link to his installation guide is here: <https://snort.org/documents/snort-3-1-0-0-on-ubuntu-18-20>

Chapter 17: Network Intrusion Detection

In this chapter, students will be focusing on the IPS virtual machine, and installing one of two open-source network intrusion detection software suites: Snort or Suricata. Both Snort and Suricata are two sides of the same coin, and share a lot of history.

Snort was originally written in 1998 as a clone of `tcpdump`, a very old, and very useful packet sniffing program. The creator of Snort, Martin Roesch, realized that the software could be configured with signatures (referred to as "rules") that could trigger alerts/notifications when network traffic matching certain patterns is observed. From there, the rest is history. In January 2021, Snort version 3 (Sometimes referred to as *Snort++*) was released after 14 years in development. It sports a host of new features, bringing it more in line with features that many competing network intrusion detection products (both closed and open-source) already enjoy.

Suricata was originally released publicly in 2010 as an alternative to Snort. It sports a variety of new features, along with compatibility with Snort 2.x rules. Since then, the project has grown steadily, with a robust collection of features that have truly allowed it to stand outside of Snort's shadow.

My Software Can Beat Your Software

So, which is the better software? Snort3 or Suricata? Let me give you the short answer: **Use whichever one you want.** The only reason we're using either of these solutions is for two purposes: bridging the IPS1 and IPS2 network segments together, and serving as a proof of concept for logging security events to our SIEM VM. Most of the advanced features that either suite has really don't make much of a difference for our use case.

If you work for a place that uses Cisco network security products, I would recommend learning Snort 3. Cisco has owned Sourcefire (and Snort) since 2013, and they have a habit of integrating their acquisitions into their product lines. So, if you do network security work, and your org uses a lot of Cisco network security products, it would make sense for you to learn Snort 3.

In almost any other circumstance, I recommend using and learning Suricata. As of right now, Suricata has been around longer than Snort 3. While technically the Snort project as a whole has been around *a lot longer* than Suricata, some of the feature that Snort 3 sports have been mainstays in Suricata for the better part of a decade. In general, new software means new bugs, and new quirks, and Snort 3, in spite of its long development history, is very new.

Additionally, Suricata's documentation is very well put together, and regularly updated for major releases. Finally, I've reached out to community members and submitted bugs (admittedly trivial bugs), and I found the community to be very responsive.

In any case, if you're curious about the features of each product, here are some links to help you out:

Suricata features: <https://suricata-ids.org/features/all-features/>

Snort features: <https://snort.org/snort3>

17.1 Making a Choice

In order to make what was a pretty labor-intensive and very manual process much easier, I've provided two scripts for students to use: Autosnort3 and Autosuricata. As the names imply, they automate a lot of the tasks associated with compiling, installing, and configuring either Snort3 or Suricata. In general, both scripts perform the following tasks:

- Installs prerequisites required to compile and run the IDS/IPS software
- Compiles the IDS/IPS software
- Installs an application that can be used to download new rules/signatures
- Configures the IDS/IPS for inline mode operation via AFPACKET bridging
- Installs a systemd service file for persistence in order for students to start/stop the service as required, and automatically start the service on system boot

If students would like to install Snort3, jump to section 17.2 (below). Otherwise, for Suricata, jump to [section 17.3](#) (p. 988)

Note: Seeing as how this chapter is after both of the routing and remote access chapters (for hosted and/or bare-metal hypervisor lab deployments), I'm going to assume that students have SSH access to all of their virtual machines – SIEM, IPS, and Kali – As well as the bastion host system (if using a bare-metal hypervisor). I'm also going to assume that if I tell you a task requires root permission or to "become the root user", you either enabled SSH access as the root user, or understand to use sudo to execute commands as the root user.

Take A Snapshot, It'll Last Longer

One more note before we get started here. These scripts install a lot of stuff and make a lot of configuration changes to the IPS VM. ***I would recommend taking a snapshot of the IPS virtual machine before installing either Snort3 or Suricata.*** Another thing you might consider if you want to experiment with both IDS/IPS software solutions (and have a bit of extra disk space to store snapshots) is taking a snapshot after the installation script finishes as well, then reverting to the snapshot, and installing the the other IDS/IPS software. Confused? Think about it like this

- Take a snapshot prior to installing Snort3 or Suricata.
- Install Snort3. Take a snapshot when the script finishes.
- Revert the snapshot to the point prior to installing Snort3, and install Suricata instead. Take a snapshot when the process completes.

You'll now be able to switch between Snort3 and Suricata IDS/IPS software to experiment with as you desire, by reverting to different snapshots.

17.2 Installing Snort3 (via Autosnort3)

SSH into the IPS VM, and become the `root` user. Afterwards, run the following commands:

```
git clone https://github.com/da667/Autosnort3
cd ~/Autosnort3/Ubuntu/AVATAR
```

Using their preferred text editor (e.g., `vi`) students will need open `full_autosnort.conf`, and enter the network interfaces attached to the IPS1 and IPS2 segments on line 12 (replacing the value `snort_iface_1=eth1`, with `snort_iface_1=[IPS1 interface name]`) and line 20 (replacing the value `snort_iface_1=eth2` with `snort_iface_2=[IPS2 interface name]`).

Note: Remember during the hypervisor setup guide when you were installing Ubuntu Server on the IPS VM and you had to cross-correlate which network interfaces were connected to the IPS1 and IPS2 network segments on the Network Setup screen? Remember how I recommended documenting configuration details for your virtual machines so you have that information to refer back to later? If you didn't do either of those things, now is as good a time as any.

The quickest way for you to determine which interface names need to be on lines 12 and 20 would be to run the `ip -br addr` command, ignore the `lo` interface, and look at the interfaces that **do not** have an IP address assigned. Usually, these interfaces will be named `ens[xxx]`, `enp[xx]s[xx]`, or `eth[xxx]`— such as `ens192` and `ens224`, `ens33` and `ens34`, `enp0s8` and `enp0s9`, or `eth1` and `eth2`. The interface names and number may vary, but it's very important that they are properly identified!

On line 32, students will need to enter a valid oinkcode from `snort.org`. Recall back in chapter 1, [section 1.5.5](#) (p. 26), students were advised to register an account on `snort.org`. If you haven't already, register an account, and when you've finished the registration process, log in, and click

on the e-mail address you used to register an account in the upper-right corner of the page. This brings you to your account information. On the left, click on the menu option labeled *Oinkcode*, copy the code in the middle of the screen, and paste it (or carefully type it) into the `o_code=` field on line 32. After students have successfully configured `snort_iface_1`, `snort_iface_2` and `o_code`, save the `full_autosnort.conf` file, and exit.

Note: As always, be aware that *software is subject to change*. This also includes websites. The `snort.org` site may change arbitrarily at any time. The key take-away is that you need to register an account to `snort.org`, and afterwards, you'll need to locate and copy your oinkcode.

The next thing we need to do is run a couple of commands to set the environment variables `http_proxy` and `https_proxy`. As the root user, run the following commands:

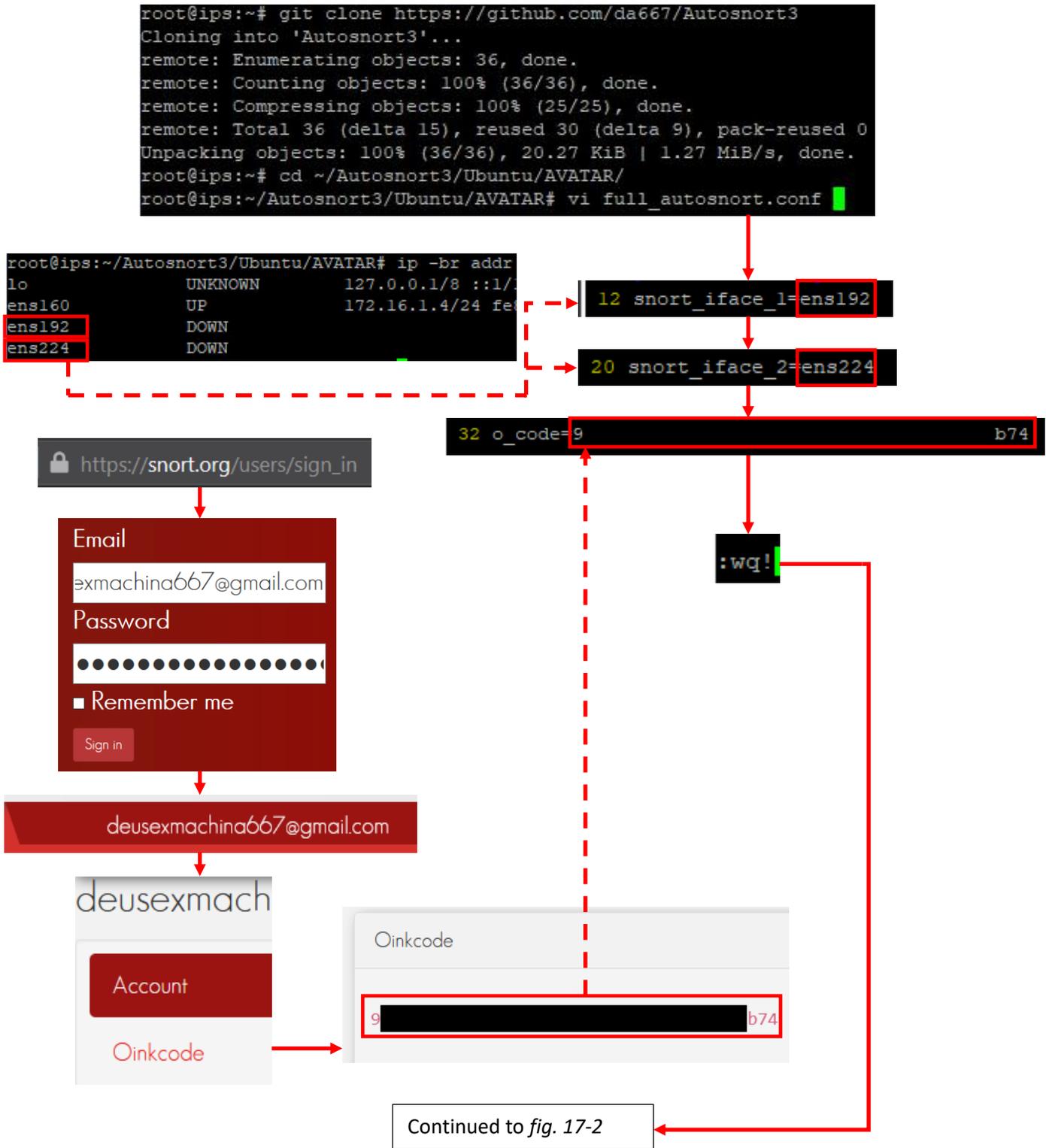
```
export http_proxy=http://172.16.1.1:3128
export https_proxy=
```

These commands are important to let the command-line session know that there is a squid proxy installed on our network that only handles HTTP traffic, and it is located on port TCP/3128 at the IP address of the pfSense VM's LAN interface. If students are using an alternate IP address layout, modify the IP address in the `http_proxy` variable to reflect your lab network. Make sure that the `https_proxy` variable is set, and that the value is blank. ***It is very important that these variables are set correctly.*** If they are not set correctly, the script may appear to "hang" or may fail unexpectedly.

Finally, we need to run the file `autosnort3-Ubuntu.sh` as the root user. The easiest way to do this is to run the following command as the root user:

```
bash autosnort3-Ubuntu.sh
```

That's it. Hard part over! The script will print periodic updates to the terminal window to let students know what it is doing while it's running. Please be aware, that Snort3, and some of its components will take a considerable amount of time to download, install and compile. During my testing it took anywhere from 45 minutes to an hour and a half for the script to complete all of its tasks and reboot, so be patient.



17-1: To run Autosnort3, students should SSH into the IPS VM and become the root user. Download the latest copy of Autosnort3 from github.com, change directories into ~/Autosnort3/Ubuntu/AVATAR, and open the full_autosnort.conf file. Fill out lines 12 and 20 with the network interfaces attached to the IPS1 and IPS2 network segments, and line 32 with the oinkcode recovered from snort.org. When finished, save the file.

Continued from *fig. 17-1*

```
root@ips:~/Autosnort3/Ubuntu/AVATAR# export http_proxy=http://172.16.1.1:3128
root@ips:~/Autosnort3/Ubuntu/AVATAR# export https_proxy=
root@ips:~/Autosnort3/Ubuntu/AVATAR# bash autosnort3-Ubuntu.sh
[*] Checking for config file..
[*] Found config file.
[*] Checking for root privs..
[*] We are root.
[*] Performing apt-get update and upgrade (May take a while if this is a fresh install)..
```

A few
moments
later...

```
[*] snort user creation successfully completed.
[*] Tightening permissions to /var/log/snort..
[*] Found snort3 systemd service script. Configuring..
[*] snort3.service installation successfully completed.
[*] Location: /etc/systemd/system/snort3.service
[*] snortd.service enable successfully completed.
[*] Rebooting now..
[*] The log file for autosnort is located at: /var/log/autosnort3_install.log
[*] We're all done here. Have a nice day.
```

17-2: With `full_autosnort.conf` modified, be sure to set the `https_proxy` variable to blank/empty string, and the `http_proxy` variable to `http://[pfSense LAN interface]:3128`. Finally, run `autosnort3-Ubuntu.sh` as the root user with the command: `bash autosnort3-Ubuntu.sh`. The script will take anywhere from 45 minutes to 1.5 hours to finish running, so be patient. When the script is done, the IPS VM will reboot.

17.2.1 Confirming Autosnort3 success

After the IPS VM has rebooted, SSH back in. Additionally, ensure that the Kali and Metasploitable 2 virtual machines are powered on. On the IPS VM, run the following command:

```
systemctl status snort3.service
```

This command queries systemd for the status of the snort3 service. Look for the output:

```
Active: active (running)
```

Congratulations, this means that snort3 is installed, and the service is running. The next task is to confirm that the IPS1 and IP2 network segments have been bridged. The simplest way for us to do that is to attempt to connect to the web server on the Metasploitable 2 VM from the Kali VM. Open a terminal session on the kali VM, and run

```
curl 172.16.2.3
```

If students are using an alternate network address configuration for the IPS network segments, substitute 172.16.2.3 with the IP address of the metasploitable 2 VM accordingly. If working correctly, students should get HTML output printed to the console that begins with:

```
<html><head><title>Metasploitable2 - Linux</title>
```

There is a bunch of other output that gets displayed (including a huge, obnoxious metasploitable 2 ascii banner), but if you see this, it means that the networking bridge is working correctly, and snort is bridging the two network segments together. Let's run a quick experiment. On the IPS VM, as the root user (or via sudo) run:

```
systemctl stop snort3.service
```

Then in the Kali VM, try to run:

```
curl 172.16.2.3
```

The command should appear to hang and/or fail with the error:

```
No route to host.
```

To start the snort service again, with root permissions on the IPS VM, run:

```
systemctl start snort3.service
```

Then after a few moments, try to run the `curl` command on the Kali VM to confirm network connectivity to the metasploitable 2 VM is available once more. If the snort3 service is showing a status other than active, or the `curl` command still fails in spite of the snort3 status being active, then that will definitely involve some troubleshooting to fix. We'll cover general troubleshooting for both Snort3 and Suricata in section 17.4 a little bit later in this chapter.

17.3 Installing Suricata (via Autosuricata)

The process for using Autosuricata is practically identical to Autosnort, covered in section 17.2 above. The only major difference between Autosnort and Autosuricata is that Autosuricata does not require an oinkcode (Autosuricata uses Proofpoint's free Emerging Threats rules). With that being said, here is a run-down of the steps to install Suricata, via Autosuricata:

Begin by establishing an SSH connection to the IPS VM and becoming the root user. As the root user, run the following commands:

```
git clone https://github.com/da667/Autosuricata
cd ~/Autosuricata/AutoSuricata-Deb/AVATAR
vi full_autosuricata.conf
```

Using vi (or your favorite text editor), modify the values `suricata_iface_1` and `suricata_iface_2` (on lines 12 and 20) in `full_autosuricata.conf` to the names of the network interfaces on the IPS VM attached to the IPS1 and IPS2 network segments, then save the file.

Note: If you documented this information when you first created the IPS VM, this section will be much easier. The quickest way for you to determine which interface names need to be on lines 12 and 20 would be to run the `ip -br addr` command, ignore the `lo` interface, and look at the interfaces that do not have an IP address assigned. Usually, these interfaces will be named `ens[xxx]`, `enp[xx]s[xx]`, or `eth[xxx]`— such as `ens192` and `ens224`, `ens33` and `ens34`, `enp0s8` and `enp0s9`, or `eth1` and `eth2`. The interface names and number may vary, but it's very important that they are properly identified!

Finally, as root, run the commands:

```
export http_proxy=http://172.16.1.1:3128
export https_proxy=
bash autosuricata-deb-AVATAR.sh
```

The `export` commands are important to let the command-line session know that there is a squid proxy installed on our network that only handles HTTP traffic, and it is located on port TCP/3128 at the IP address of the pfSense VM's LAN interface. If students are using an alternate IP address layout, modify the IP address in the `http_proxy` variable to reflect your lab network. Make sure that the `https_proxy` variable is set, and that the value is blank. ***It is very important that these variables are set correctly.*** If they are not set correctly, the script may appear to "hang" or may fail unexpectedly. Finally, we tell the bash shell interpreter to run the `autosuricata` script.

While `autosuricata` is running, it will print periodic updates to the terminal window to let students know what it is doing while it's running. Please be aware, that Suricata, and some of its components will take a considerable amount of time to download, install and compile. During my testing, it took anywhere from 45 minutes to an hour and a half for the script to complete all of its tasks and reboot, so be patient.

```
root@ips:~# git clone https://github.com/da667/Autosuricata
Cloning into 'Autosuricata'...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 86 (delta 4), reused 11 (delta 4), pack-reused 71
Unpacking objects: 100% (86/86), 28.60 KiB | 552.00 KiB/s, done.
root@ips:~# cd ~/Autosuricata/AutoSuricata-Deb/AVATAR/
root@ips:~/Autosuricata/AutoSuricata-Deb/AVATAR# vi full_autosuricata.conf
```

```
root@ips:~/Autosnort3/Ubuntu/AVATAR# ip -br addr
lo                UNKNOWN    127.0.0.1/8 ::1/128
ens160            UP         172.16.1.4/24 fe80::200:1:1:1/64
ens192            DOWN
ens224            DOWN
```

```
12 suricata_iface_1=ens192
```

```
20 suricata_iface_2=ens224
```

```
:wq!
```

```
root@ips:~/Autosuricata/AutoSuricata-Deb/AVATAR# export http_proxy=http://172.16.1.1:3128
root@ips:~/Autosuricata/AutoSuricata-Deb/AVATAR# export https_proxy=
root@ips:~/Autosuricata/AutoSuricata-Deb/AVATAR# bash autosuricata-deb-AVATAR.sh
[*] Checking for config file..
[*] Found config file.
[*] Checking for root privs..
[*] We are root.
[*] Performing apt-get update and upgrade (May take a while if this is a fresh install)..
```

A few
moments
later...

```
[*] modifying permissions to allow the suricata user and group access to rules, config files, etc.
[*] /usr/local/var/run/suricata already exists.
[*] suricata user ownership of /usr/local/var/lib/suricata successfully completed.
[*] suricata user ownership of /usr/local/etc/suricata successfully completed.
[*] Rebooting now..
[*] The log file for autosuricata is located at: /var/log/autosuricata_install.log
```

17-4: The process for using Autosuricata to install Suricata is more or less identical to the process for installing snort via Autosnort 3: as the root user, clone the Autosuricata github project, modify the full_autosuricata.conf file (no oinkcode needed), set the http_proxy and https_proxy variables, run the autosuricata-deb-AVATAR.sh shell script, and wait for the IPS VM to reboot.

17.3.1 Confirming Autosuricata success

After the IPS VM has rebooted, SSH back in. Additionally, ensure that the Kali and Metasploitable 2 virtual machines are powered on. On the IPS VM, run the following command:

```
systemctl status suricata.service
```

This command queries systemd for the status of the suricata service. Look for the output:

```
Active: active (running)
```

Congratulations, this means that suricata is installed, and the service is running. The next task is to confirm that the IPS1 and IP2 network segments have been bridged. The simplest way for us to do that is to attempt to connect to the web server on the Metasploitable 2 VM from the Kali VM. Open a terminal session on the kali VM, and run

```
curl 172.16.2.3
```

If students are using an alternate network address configuration for the IPS network segments, substitute 172.16.2.3 with the IP address of the metasploitable 2 VM accordingly. If working correctly, students should get HTML output printed to the console that begins with:

```
<html><head><title>Metasploitable2 - Linux</title>
```

There is a bunch of other output that gets displayed (including a huge, obnoxious metasploitable 2 ascii banner), but if you see this, it means that the networking bridge is working correctly, and snort is bridging the two network segments together. Let's run a quick experiment. On the IPS VM, as the root user (or via sudo) run:

```
systemctl stop suricata.service
```

Then in the Kali VM, try to run:

```
curl 172.16.2.3
```

The command should appear to hang and/or fail with the error:

```
No route to host.
```

To start the snort service again, with root permissions on the IPS VM, run:

```
systemctl start suricata.service
```

Then after a few moments, try to run the `curl` command on the Kali VM to confirm network connectivity to the metasploitable 2 VM is available once more. If the `suricata` service is showing a status other than active, or the `curl` command still fails in spite of the `suricata` status being active, then that will definitely involve some troubleshooting to fix. We'll cover general troubleshooting for both Snort3 and Suricata in section 17.4.

17.4 Troubleshooting Snort and Suricata problems

Below is a collection of troubleshooting considerations to think about if things are not working as intended. Hopefully these troubleshooting recommendations will lead students to a favorable conclusion.

- Troubleshooting Autosnort or Autosuricata:
 - Does the IPS VM have internet connectivity?
 - `nslookup google.com`
 - `curl -I https://www.google.com`
 - Run the `export` command
 - Are the variables `http_proxy` and `https_proxy` defined? Is `http_proxy` set to `http://[pfSense LAN interface]:3128`?
 - Is the squid proxy service installed, and configured on the pfSense VM? Installing and configuring the squid proxy service was covered in Chapter 14.
 - Is the firewall policy correctly configured on the pfSense LAN interface? Firewall policies on the LAN interface for hosted and/or bare-metal hypervisors were also covered in Chapter 14.
 - Is the `full_autosnort.conf` or `full_autosuricata.conf` config file filled out correctly?
 - Line 12 and Line 20 need to be filled out with the network interfaces attached to the IPS1 and the IPS2 network segments.
 - Remember, `ip -br addr` can be used to figure out which network interfaces to use. These interfaces will be marked as `DOWN`, and will not have an IP address assigned.
 - Autosnort requires an oinkcode defined on line 32 of the `full_autosnort.conf` file.
 - Register an account on `snort.org`, log in, and recover your oinkcode.
 - Both Autosnort and Autosuricata log the output of every single command that the script runs. If either of the scripts fail, check out `/var/log/autosnort3_install.log` or `/var/log/autosuricata_install.log`. These logs are extremely valuable in troubleshooting why the script failed.
 - I noticed with Autosnort3 in particular that most common failure conditions come with failing to download `libdaq`, `snort3` itself, or the latest `snortrules-snapshot` with `pulledpork.pl`
 - Usually, the `autosnort3_install.log` will reveal that an HTTP 500 series error when downloading `libdaq`, or `snort3` if the download fails.
 - With `pulledpork`, this commonly would resort with HTTP error code 422 Unprocessable entity if it fails to download the `snortrules` snapshot.
 - Sometimes, this can be an indication that the oinkcode is invalid. Double check that the oinkcode was entered into the `full_autosnort.conf` file correctly. All oinkcodes (at least as of mine writing this) are 40-character alphanumeric strings.
 - If students experience either the HTTP 500 series or 422 error code, it usually indicates that Cisco's infrastructure serving these files is over capacity. The only thing I can recommend is to attempt running the script again.

- Troubleshooting snort or suricata:
 - Are the interfaces configured in `full_autosnort.conf/full_autosuricata.conf` attached to the correct network segments?
 - Every hypervisor covered in the hypervisor setup guides has a different name for its network segments and virtual switches. Refer back to the hypervisor setup guide you used to set up the IPS virtual machine, and confirm the network interfaces are attached to the correct network segments/virtual switches.
 - Are required network features enabled (promiscuous mode, MAC spoofing, etc.)?
 - Certain hypervisors require extra network configuration options to be enabled in order for Snort and/or Suricata to function properly. Make absolutely sure that these additional network features are enabled. Students should check the chapter review portion of their chosen hypervisor setup guide.
 - Is the `snort3.service` or `suricatad.service` running?
 - Run `systemctl status suricatad.service` for suricata, and `systemctl status snort3.service` for snort to see if the service is in the active status.
 - Is the snort or suricata process running?
 - Run the command `ps -ef | grep snort` for snort. Check for the following output:
 - `/usr/local/bin/snort -c /usr/local/etc/snort/snort.lua -D -u snort -g snort -l /var/log/snort -m 0x1b --create-pidfile --plugin-path=/usr/local/lib/snort_extra -s 65535 -k none -Q`
 - Run the command `ps -ef | grep suricata` for suricata. Check for the following output:
 - `/usr/local/bin/suricata -D -c /usr/local/etc/suricata/suricata.yaml --af-packet --user=suricata`
 - If the snort or suricata process/service isn't running, check `/var/log/syslog` for clues.
 - `cat /var/log/syslog | grep snort`
 - `cat /var/log/syslog | grep suricata`
 - Are the Kali and Metasploitable 2 VMs powered on and assigned the correct IP addresses?
 - Confirm that the Kali VM and Metasploitable 2 VM are assigned the correct IP addresses. Kali should be assigned the IP address 172.16.2.2, while Metasploitable 2 should be assigned the IP address 172.16.2.3. Of course, if students are running an alternate IP address range for the IPS network segments other than 172.16.2.0/24, adjust the IP address assignments and the IP address for the `curl` command accordingly.
 - On the Kali VM, run `ip -br addr` to verify the IP address.
 - On Metasploitable 2, run `ifconfig -a`.
 - If the metasploitable 2 VM does not have an IP address assigned to the `eth0` interface, confirm that the pfSense VM has a DHCP allocation configured for the Metasploitable 2 VM's MAC address on the OPT1 interface. Instructions on how to configure a static DHCP allocation for the Metasploitable 2 VM were covered in the hypervisor setup guides, while students were configuring the Metasploitable 2 VM.
 - If there is a DHCP mapping configured for the Metasploitable 2 VM's MAC address, may try running `dhclient -i eth0` as the root user, or rebooting the virtual machine.

17.5 Chapter Review

In this chapter we covered installing Snort3 or Suricata network intrusion detection software in order to inspect network traffic in the IPS network, bridge the IPS1 and IPS2 network segments together, and serve as a fail-close network bridge. Students were provided instructions on how to acquire either Autosnort or Autosuricata, configure the script, execute it, then test the AFPACKET bridge provided by Snort or Suricata to confirm the software is working as intended. Additionally, students were provided with a collection of network troubleshooting considerations that can be used to hopefully resolve any problems they had with the Autosnort/Autosuricata scripts, or the Snort/Suricata software itself.

One more chapter completed! We're getting close to the end of the road, now. Students should to proceed to chapter 18 in order to set up Splunk on the SIEM virtual machine as well as forwarders on the IPS VM so that we can forward network traffic logs to the SIEM VM for review.

Chapter 18 Patch Notes

-A lot has changed with Splunk's licensing practices since the first book was published. Turns out, developer licensing is a lot harder to acquire now, basically requiring users to use a company e-mail address for their Splunk account, and for their company to be existing Splunk customers

-It's not tenable for me to tell students to work on acquiring a developer license. The only responsible thing I can tell students to do is either deal with the 500MB per day limit for Splunk free, or recommend possible alternatives.

-Unfortunately, configuring a Splunk indexer/searchhead for Splunk free disables some features, namely user authentication. I'm kinda disappointed that being able to configure authentication is considered a premium feature, but by sheer chance, Noah Dietrich's Snort3 installation guide provides some recommendations to set up the Splunk app behind an Apache reverse proxy. This allows us to set up HTTP basic auth, and provide some form of authentication for the Splunk web interface when using Splunk free edition. I created an extra content sidebar conversation based on his content that guides students on how to do this. The only difference between my configuration and Noah's is that my reverse proxy configuration uses self-signed SSL because HTTP basic auth over plain HTTP is pretty insecure.

- Of course, this completely optional since secure network access to the SIEM VM should more or less be handled by now, but it's an extra layer of security.

- It turns out that nikto is pretty good at generating a barrage of IDS alerts and pretty quickly at that, so I'll be using that to help students confirm that the IDS, the universal forwarder, and Splunk are working, instead of trying to use armitage. Armitage was considered old when I made the first book. At this point it borderline irresponsible to recommend using it for much of anything, especially considering that its effectively no longer maintained.

- Added a troubleshooting section to this chapter to help students with recommendations on things to do, and places to look if logs aren't making it to the SIEM VM.

Chapter 18: Setting up Splunk

As the name of this chapter implies, students will be learning how to install Splunk for their lab environment. The Splunk deployment will consist of the SIEM VM serving as both an indexer and search head, and the IPS VM configured to forward events from either Snort or Suricata to the SIEM VM via a universal forwarder. If you're not down on Splunk lingo, don't panic, I'll be walking students through all the steps necessary to get it all up and running.

I'll be dividing this chapter up to three major parts: Setting up Splunk on the SIEM VM, Setting up the forwarder on the IPS VM (with a subsection for Snort and one for Suricata), and a final section on how students can generate data using the Kali, Metasploitable and IPS VMs, and how to query for that generated data on the SIEM VM.

Note: I feel obligated to mention that I currently work for a company in which I make a living using Splunk for network security monitoring. However, just because I use it and prefer it doesn't mean you have to. If there is another SIEM or log management solution you prefer to use instead (e.g., ELK (Elastic Search, Logstash, Kibana/Graylog), Apache Metron, MozDef, SIEMonster, etc.), and you're confident you know well enough how to get it to work (or want to experiment for yourself), feel free to substitute Splunk with whatever suits you.

18.1 Installing Splunk on the SIEM VM

This section will show students how to download, install, and configure Splunk on the SIEM VM. Before we get started, in chapter 1, [section 1.5.5](#) (p. 26), I recommended registering an account on splunk.com. If students have not performed this task yet, please be sure to do so. An account will be required to download the Splunk Enterprise installer to the SIEM VM. Speaking of the SIEM VM, now is a really good time to make a virtual machine snapshot/checkpoint before making any major changes, just in case something unexpected occurs. When students are ready to proceed, open up an SSH session to the SIEM VM, and become the `root` user.

18.1.1 Downloading Splunk Enterprise

Upon logging in, select the *Products* menu option at the top of the web page, and in the drop-down menu that appears, select *Free Trials & Downloads*. On the next page, under the *Splunk Platform Products* section, Click the link under *Splunk Enterprise*, labeled *Download Free 60-Day Trial*. On the next page, under the *Choose Your Installation Package* table, click on *Linux*, then click the *Download Now* button to the right of the `.deb` option

On the next page, students will be subjected to the Splunk Software License Agreement. Agree to the terms of the license (without reading it, as is tradition), then click the *Start Your Download Now* button. Students will be directed to a new page with large text that reads *You're Downloading Splunk Enterprise [current version] for Linux*. The web browser of your choice will

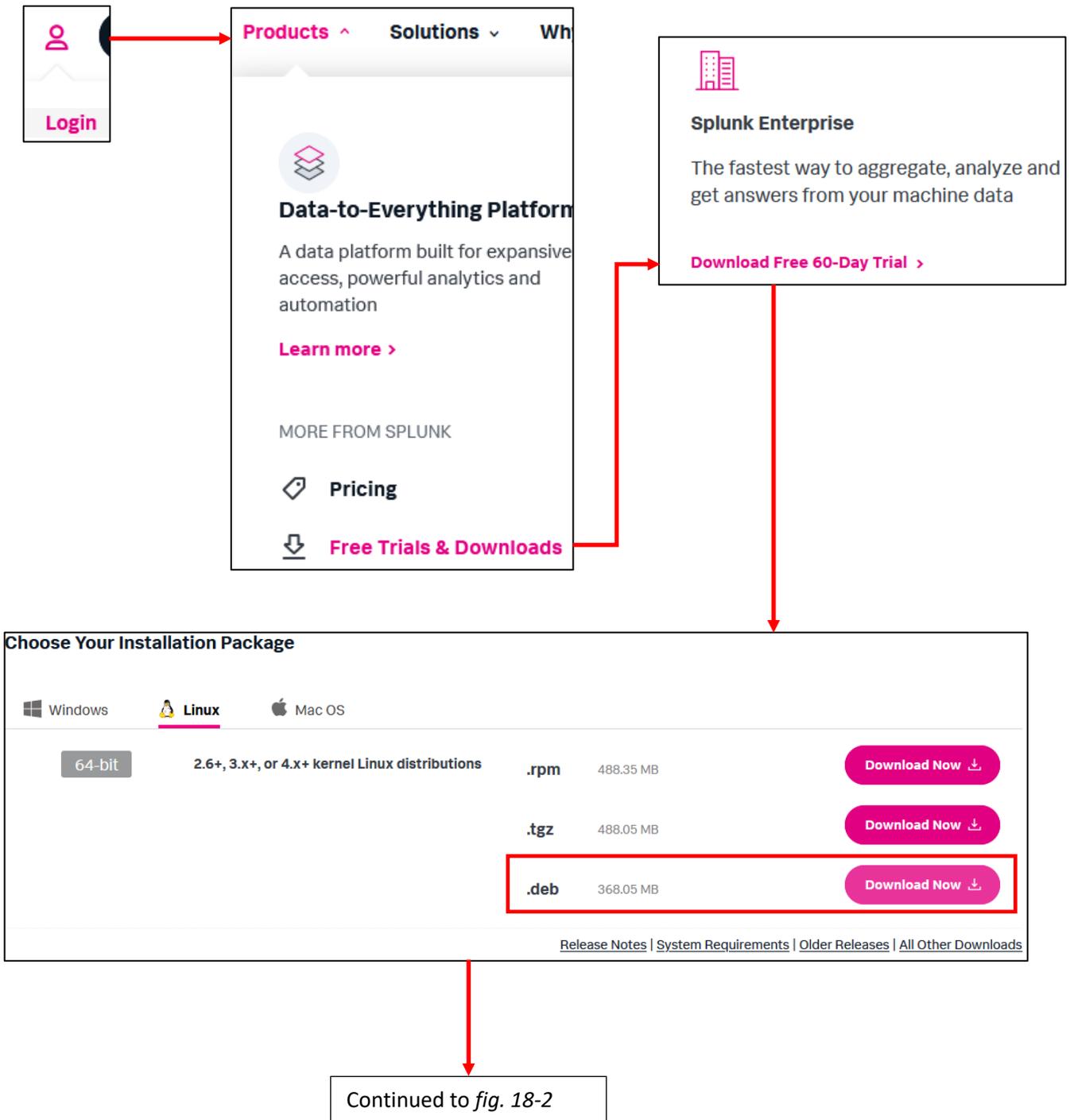
offer to download the package you requested. Hit *Cancel* on the browser download box that appears, and to the right of the text *You're Downloading Splunk Enterprise* is a small box labeled *USEFUL TOOLS*. The first bullet reads *Download via Command Line (wget)*. Click that link, and a new box appears. Part of the text in that box reads *Click [here](#) to select the entire command*. Click that link, and the entire `wget` command below will be highlighted. Students need to copy the ***entire*** `wget` command to their system's clipboard, then paste that into the terminal with the SSH session on the SIEM vm. As always, the download link is subject to change, but here is the `wget` command I used to download Splunk Enterprise:

```
wget -O splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb  
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.3&product=splunk&filename=splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb&wget=true'
```

When the download completes, the file `splunk-[current major version]-[build number]-linux2.6-amd64.deb` should be present on the system. Once again, for this lesson, the file I downloaded was named

```
splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb
```

Note: As always, software is subject to change. That includes websites. Companies like to change website layouts all the time, for seemingly arbitrary reasons. That means that the installers we need to download for the lab environment may change locations over time. Just remember: the overall goal is to get the Splunk Enterprise `.deb` package for the SIEM VM, and universal forwarder `.deb` package for the IPS VM.



18-1: Students need to download Splunk Enterprise. To do that, visit <https://www.splunk.com> and log in. Navigate to *Products* and click *Free Trials & Downloads*. On the next page, locate Splunk Enterprise and click the link *Download Free 60-day Trial* link. A new page with a table labeled *Choose Your Installation Package* appears. Click on *Linux*, then click the *Download Now* button to the right of the *.deb* option.

Continued from *fig. 18-1*

I have read, understood and hereby accept the Splunk Software License Agreement.

↓ Start Your Download Now

USEFUL TOOLS

- Download via [Command Line \(wget\)](#)
- Download via [Browser](#)
- Get data from [Forwarders](#)

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click [here](#) to select the entire command.

```
platform=linux&
version=8.1.3&
product=splunk&
filename=splunk-8.1.3-63079c59e632-2.6-amd64.deb&wget=true'
```

Undo
Cut
Copy
Paste
Delete

```
root@siem:~# wget -O splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=8.1.3&product=splunk&filename=splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb&wget=true'
```

18-2: After accept the Splunk license agreement, students will be taken to the *You're Downloading Splunk Enterprise* page, and will likely get a dialogue box pop-up from their browser asking where to download the Splunk Enterprise installation package. Click *Cancel* on that dialogue box, and focus on the box labeled *USEFUL TOOLS*, specifically the first bullet point labeled *Download via Command Line (wget)*. Click that link and a pop-up appears with a long, fully formatted wget command. Copy and paste this command into the SIEM VM SSH session to download the Splunk Enterprise .deb package directly to the SIEM VM.

18.1.2 Installing and Configuring Splunk (Part 1)

With the .deb package downloaded on the SIEM VM, run the following commands as the root user:

```
dpkg -i splunk-*.deb
chown -R splunk:splunk /opt/splunk
/opt/splunk/bin/splunk start --answer=yes --accept-license
```

This small sequence of commands using `dpkg`, the Debian/Ubuntu package manager to install the splunk enterprise .deb file. All of the Splunk components and files are installed to `/opt/splunk`. As a part of the installation process, Splunk Enterprise creates a system user named `splunk`. The `chown` command is used to make sure that all of the files in `/opt/splunk` are owned by the `splunk` user and group. This is necessary to start and stop Splunk and its services. Finally, the command `/opt/splunk/bin/splunk start --answer=yes --accept-license` is used to start up Splunk's services and do some of the initial setup tasks. The extra command options answer yes to any questions the first-time setup process asks, and also skips having to read and accept the end-user license agreement (as is tradition). However, there is one task that cannot be skipped during the first-time setup process, and that is creating an administrative user for the Splunk web interface. Students will be prompted to enter a username and password. **These credentials will be the username and password students will need to log into the Splunk web interface later, so as with all of your important credentials for the lab environment so far, store them some place safe.**

Next, run the following commands with root permissions:

```
/opt/splunk/bin/splunk stop
/opt/splunk/bin/splunk enable boot-start -systemd-managed 1
chown -R splunk:splunk /opt/splunk
systemctl start Splunkd.service
systemctl status Splunkd.service
```

This set of commands is to ensure that the Splunk services on the SIEM VM start automatically whenever the system is rebooted for any reason. We begin by stopping the current running instance of Splunk Enterprise in order to enable the `boot-start` settings, and tell Splunk that we want to integrate it with `systemd` to handle service control. Unfortunately, this sometimes has the side-effect of changing the file permissions of a host of files in `/opt/splunk` to where they are owned by the root user, so we have to use `chown` again to reset their permissions and ensure the `splunk` user/group own everything in `/opt/splunk` again. Finally, we use `systemctl` to both start the `Splunkd` service, then confirm it is in the active (running) state.

```

root@siem:~# dpkg -i splunk-*.deb
Selecting previously unselected package splunk.
(Reading database ... 144109 files and directories currently installed.)
Preparing to unpack splunk-8.1.3-63079c59e632-linux-2.6-amd64.deb ...
Unpacking splunk (8.1.3) ...
Setting up splunk (8.1.3) ...
complete
root@siem:~# chown -R splunk:splunk /opt/splunk/
root@siem:~# /opt/splunk/bin/splunk start --answer-yes --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: ayy
Password must contain at least:
    * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Waiting for web server at http://127.0.0.1:8000 to be available... Done

If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://siem:8000
root@siem:~# /opt/splunk/bin/splunk stop
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.
...
Stopping splunk helpers...

Done.
root@siem:~# /opt/splunk/bin/splunk enable boot-start -systemd-managed 1
Important: splunk will start under systemd as user: splunk
Systemd unit file installed at /etc/systemd/system/Splunkd.service.
Configured as systemd managed service.
root@siem:~# chown -R splunk:splunk /opt/splunk/
root@siem:~# systemctl start Splunkd.service
root@siem:~# systemctl status Splunkd.service
● Splunkd.service - Systemd service file for Splunk, generated by 'splunk enable boot-start'
   Loaded: loaded (/etc/systemd/system/Splunkd.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2021-04-22 00:02:03 UTC; 9s ago

```

18-3: Use the dpkg command to install Splunk Enterprise. When finished, run chown to grant ownership of everything in /opt/splunk to the splunk user. Next, run /opt/splunk/bin/splunk start --answer-yes --accept-license to perform Splunk's first-time setup tasks, including creating a username and password to access the web interface. Then use the stop variant of the command to stop Splunk. Once splunk has shut down, run the splunk command again with the arguments enable boot-start -systemd-managed 1. After running these commands, students will have to run chown again to grant ownership of /opt/splunk back to the splunk users. Finally, users can use systemctl start Splunkd.service, and systemctl status Splunkd.service to start Splunk, then confirm the service is running.

18.1.3 Installing and Configuring Splunk Enterprise (Part 2)

Students can open a new tab on their web browser and attempt to log in to Splunk on the SIEM VM by entering:

```
http://[SIEM IP address]:8000
```

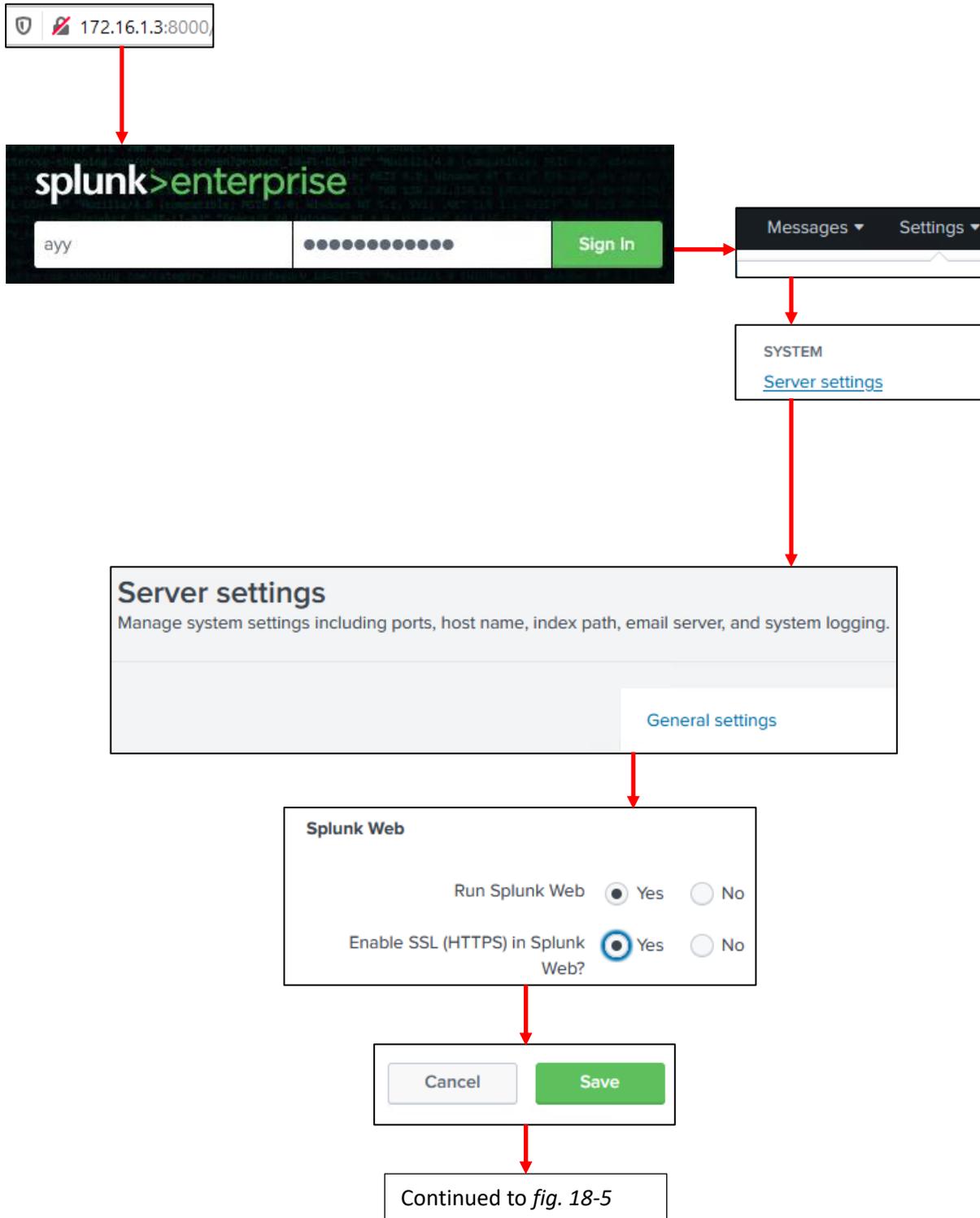
Replacing [SIEM IP address] With the IP address of the SIEM VM students assigned in their lab environment (e.g., `http://172.16.1.3:8000`). If a prompt for a username and password shows up, then that means the Splunk app was successfully installed, and is running correctly. The next step is to log in using the username and password configured during the installation process on the SIEM VM SSH session. Once students have successfully managed to log in, there are three tasks that students will need to perform: Enable HTTPS for the web interface, switch to Splunk's Free licensing, and set up a listener to accept logs forwarded from the IPS VM.

18.1.3.1 Enabling SSL on Splunk Web

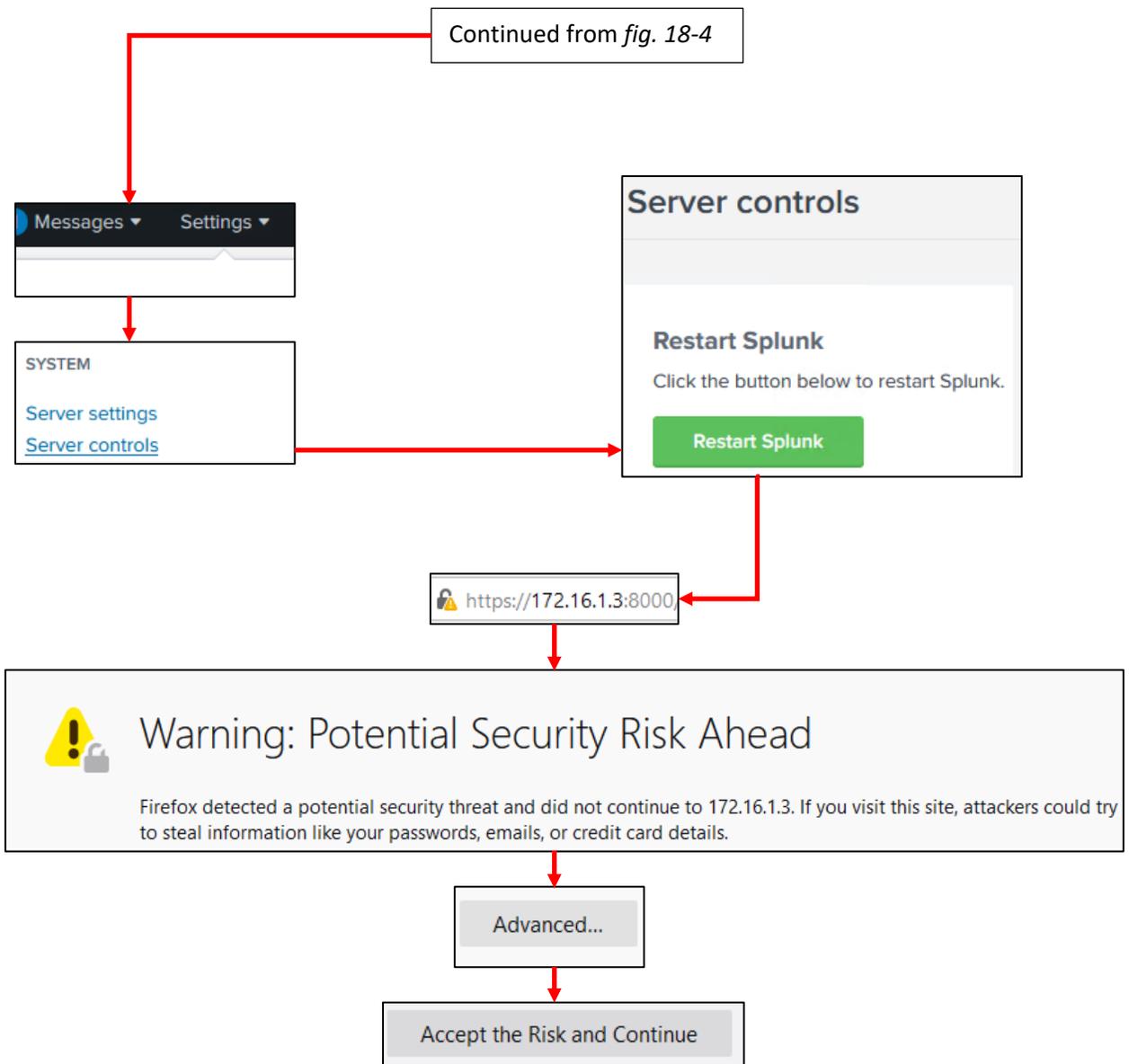
In the upper right corner of the web page are a host of options. Click on the menu option labeled *Settings*, and in the drop-down menu, select *Server Settings* under *System*. On the *Server Settings* page, select *General Settings*. Under the Splunk Web section, there is an option labeled Enable SSL (HTTPS) in Splunk Web? Select the Yes radio button, then scroll to the bottom of the page, and click the *Save* button. In order for this change to take effect, Splunk must be restarted. There are two ways to accomplish this: Reboot the SIEM VM, or navigate to *Settings > Server Controls*, then click the *Restart Splunk* button. A dialogue box will appear to confirm your choice, click OK to proceed. Wait about 5 minutes, then in the URL bar of your web browser type in:

```
https://[SIEM IP address]:8000
```

Once again, students will need to substitute and enter the IP address of the SIEM VM for their lab environment. Not unlike the first time students connected to the pfSense webConfigurator, students should be greeted with some form of warning from their web browser, suddenly advising them not to visit the Splunk web interface. And just like before, its because web browsers are extremely paranoid about self-signed SSL certificates. Select the necessary options on your web browser's warning dialogue to continue connected. For example, on Mozilla Firefox and its variants, its *Advanced > Accept the Risk and Continue*.



18-4: Log in to the Splunk web interface by entering `http://[SIEM IP address]:8000` in the web browser's URL bar. Students should log in using the username and password defined during the first-time setup process. Upon logging in, in the upper right corner, select *Server Settings > General Settings*, and on the *General Settings* page under *Splunk Web*, Set *Enable SSL (HTTPS) in Splunk Web?* To *Yes*, then click the *Save* button.



18-5: After enabling HTTPS access, navigate to *Settings > Server controls*, and click the *Restart Splunk* button. After about five minutes or so, enter `https://[SIEM IP address]:8000` into the browser's URL bar. Students should be aware that when they do this, most modern web browsers *really* don't like self-signed SSL certificates and will display all sorts of scary warnings. Just like they did when we first logged in to the pfSense webConfigurator. As usual, ignore these warnings and log back in to the Splunk web interface.

18.1.3.2 Configuring a Receiver

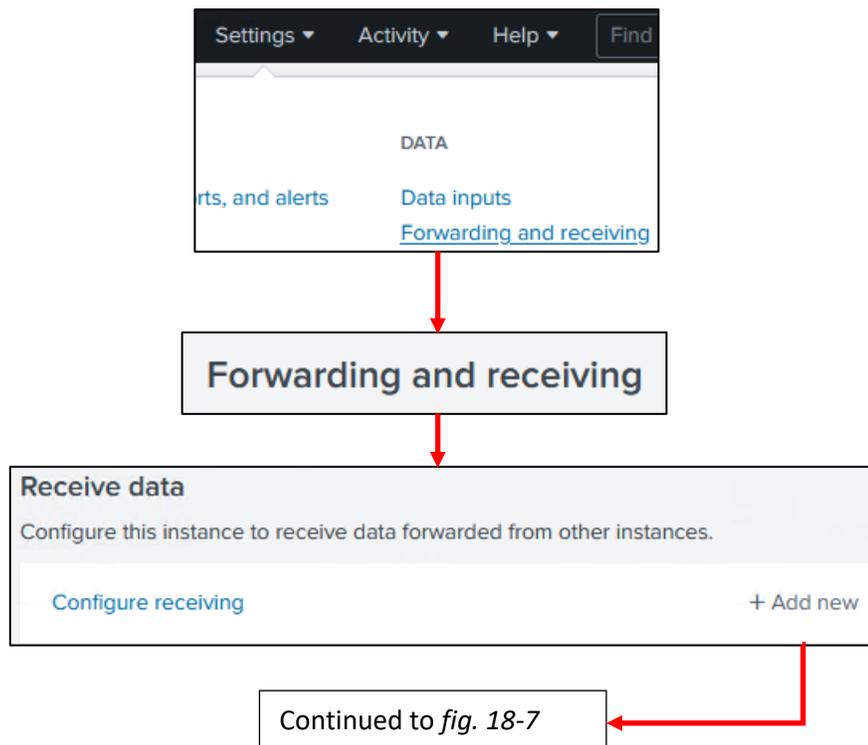
When students are finished enabling HTTPS access, log back in to the Splunk web interface. Navigate to *Settings*, in the upper right corner of the web page, then select *Forwarding and Receiving* under the *Data* section of the drop-down menu. On the *Forwarding and receiving* page, click the text *+Add new* to the right of *Configure receiving* under the *Receive data* section. On the new page that appears, enter 9997 in the input box labeled *Listen on this port*, and click the *Save* button. After clicking *Save*, students will be directed to a page labeled *Receive data*. A small table with the TCP listener on port 9997 should be listed. Under the *Status* column, the text *Enabled* should be black, confirming that our Splunk instance is listening for forwarded logs on port 9997/tcp. Students can also open an SSH session to the SIEM VM and run the command:

```
ss -a|nt4 | grep 9997
```

This command will query the SIEM VM for all listening IPv4 TCP sockets. We then pipe that output to grep and look for the string "9997". This should return output that looks something like:

```
LISTEN 0      128          0.0.0.0:9997      0.0.0.0:*
```

If this output is present, then that means the SIEM VM is listening for TCP connections on port 9997, and is ready to receive logs from the IPS VM – we'll be covering that in a little bit.



18-6: Navigate to *Settings* > *Forwarding and receiving*, then click on *+ Add new* under the *Receive data* section, to the right of *Configure receiving*.

Continued from *fig. 18-6*

Configure receiving

Set up this Splunk instance to receive data from forwarder(s).

Listen on this port *

Receive data

Forwarding and receiving » Receive data

Successfully saved "9997".

Showing 1-1 of 1 item

filter

Listen on this port	Status
9997	Enabled Disable

```
ayy@siem:~$ ss -alnt4 | grep 9997
LISTEN 0      128          0.0.0.0:9997  0.0.0.0:*
```

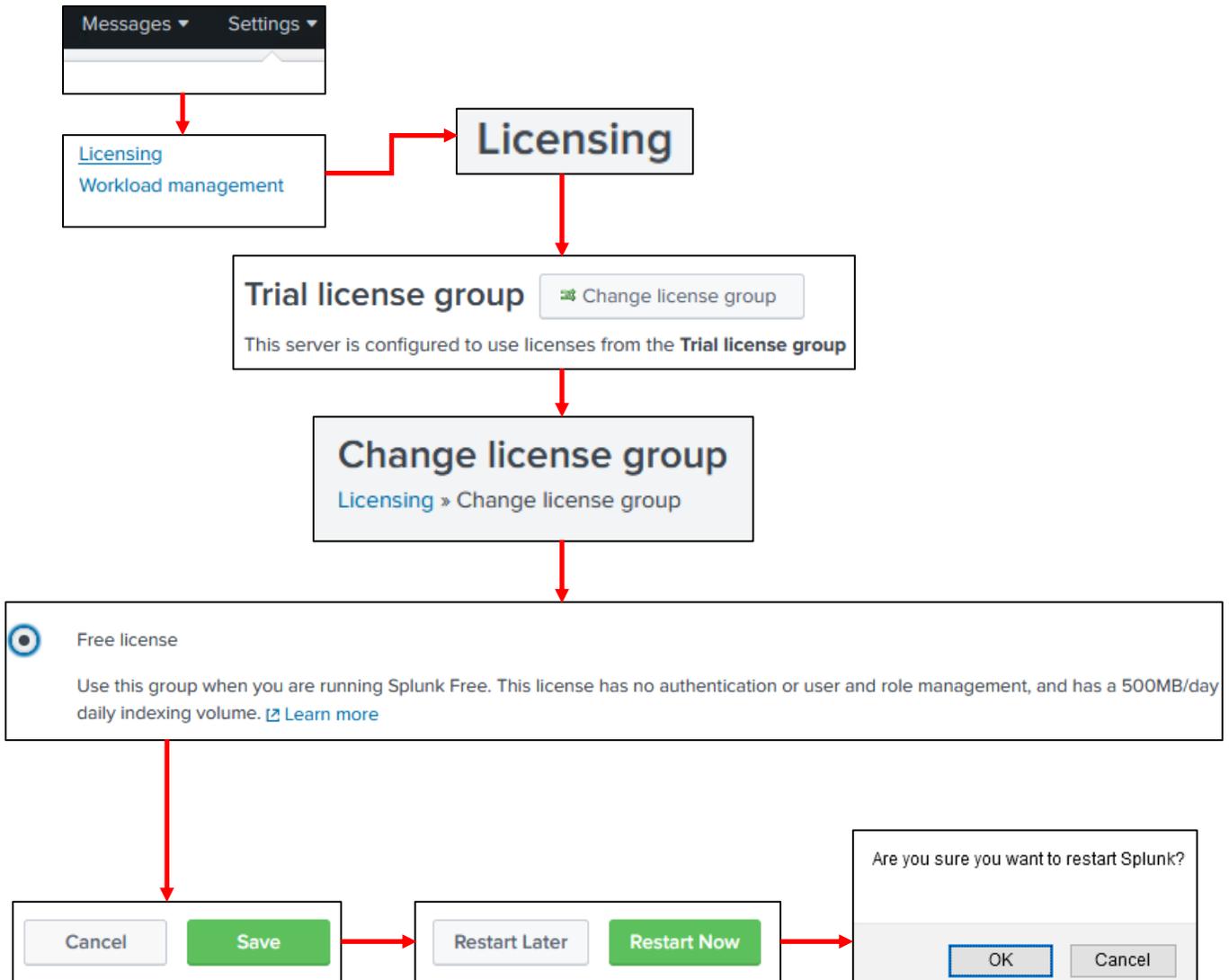
18-7: On the next page, under the *Configure receiving* section, enter 9997 into the *Listen on this port* input box, then click *Save*. Students will be directed to the *Receive data* page. If the data receiver/listener was configured correctly, there should be one entry in the table on this page. Under the *Listen on this port* column should be the entry 9997. And under the *Status* column, the text *Enabled* should appear in black, confirming that the service is listening for forwarded logs on TCP port 9997. Students may also open an SSH session to the SIEM VM, and run:

```
ss -alnt4 | grep 9997
```

To confirm that that the SIEM VM is listening on TCP port 9997 for network connections.

18.1.3.3 Switching to Splunk Free Licensing

In the *Settings* menu, select *Licensing* under the *System* section. On the *Licensing* page, click the button labeled *Change license group*. On the *Change license group* page, select the *Free license* radio button, then click *Save*. A new page appears, informing students that a restart is required to apply these changes. Click the *Restart Now* button to continue. A dialogue box will appear asking students if they're sure they wish to restart. Click the *OK* button to proceed. After a few minutes, students may refresh their web browser to regain access to the Splunk web interface.



18-8: To enable Splunk's free license mode, Navigate to *Settings > Licensing*. On the *Licensing* page, click the *Change license group* button. On the *Change license group* page, click the *Free license* radio button, then click *Save*, *Restart Now*, then finally click the *OK* button. After a few moments, refresh the web browser to regain access to the Splunk web interface.

The Price You Pay

There are some... *limitations* that come with running Splunk with free licensing. If you were paying attention to the *Free license* radio button, it lists some of the stipulations. First and foremost, regardless of whether or not you are using a free license, or the trial Splunk Enterprise license, Splunk only allows trial/free users to index up to 500MB of data per day. The next big limitation is that switching to free licensing disables authentication, users, and/or role management on the web interface. This means that anyone that knows the URL of the Splunk web interface on the SIEM VM, and has network access to the SIEM VM will be able to access Splunk and do practically whatever they want.

With regards to the data limitation, 500MB doesn't sound like a lot, but just remember that this is 500MB of text logs. When we're done here, your Splunk instance will only have one data source feeding into it: Either Snort3 JSON logs, or Suricata JSON logs. Unless you're doing a lot of heavy duty traffic analysis in your lab environment all day long, the likelihood of most students reaching this limit isn't very high, however I won't say it's impossible.

Now, as for the other limitation, it is a *slight* problem that enabling Splunk Free licensing completely disables authentication, users, and roles, but your lab environment is built to limit access to your lab virtual machines. This is why we took so much time to ensure that only the hypervisor host could access lab VMs if you're using a hosted hypervisor, or to ensure that only specifically allowed bastion hosts could access the lab environment for bare-metal hypervisors. However, there is one ingenious work-around to this problem that I've found provided by Noah Dietrich (I mentioned his documentation as a huge influence for helping me write Autosnort3 in the last chapter). We'll discuss that in the next sidebar conversation.

Now, there is another option, but it's a lot more limited now than it used to be. To make a long story short, if your employer (or maybe your college – talk to your college faculty?) are enrolled as Splunk customers, and you register an account with splunk.com using your company/college e-mail, you can attempt to request a developer license. The current link for this program is at:

https://www.splunk.com/en_us/resources/personalized-dev-test-licenses.html

Now, as I've stated before many times, this lab network is meant to be a starting point, and ultimately you control what happens to your lab network. There are other SIEM projects out there that are either free and open-source, or feature community editions that do not have some of Splunk's limitations such as: Alienvault OSSIM, Security Onion (pre-configured elastic search/logstash/kibana stack), SELKS (same as security onion, but exclusively for Suricata), Graylog open, SIEMonster, MozDef, Apache Metron, etc.

If you need access to solutions that allow more than 500MB/day, or just want try out an alternative to Splunk, considering giving these alternatives a try, instead.

Extra Content: Putting Splunk Free Behind a Reverse Proxy

Sometimes as an IT professional, you're forced to run things on your network that can only be described as "grody". You know, legacy applications that only allow 8-character alphanumeric passwords, log management platforms that consider authentication to be a paid luxury, anything ICS/SCADA, etc. You need ways to contain that grodiness. These are known as mitigating factors, mitigations, and/or work-arounds.

As I mentioned in sidebar conversation above, *The Price You Pay*, there is a way to put at least *some* form of authentication in front of Splunk free. In this section, students will configure Splunk free to listen on 127.0.0.1:8443, then install apache web server, configure it as a reverse proxy, enable basic authentication, and have it proxy connections to the Splunk web application. The benefit of you learning how to do this now is that this knowledge can be adapted to other web applications as necessary. As always, this content is entirely optional, and requires a little bit effort to pull off, so before getting started, I'd recommend taking a snapshot/checkpoint of the SIEM VM, just in case things go off the rails.

As with most of these adventures that involve installing and configuring software that alters the system's state, I recommend either opening an SSH session to the SIEM VM as the `root` user, or using `sudo su -` to become the `root` user, as practically of the commands and configuration files you'll need to edit require `root` access. First, students will need to modify the file `/opt/splunk/etc/system/local/web.conf` using their favorite text editor. In this instance, I will use the command:

```
vi /opt/splunk/etc/system/local/web.conf
```

This file should contain the following lines:

```
[settings]
enableSplunkWebSSL = 1
```

We need to append `server.socket_host = localhost` to the end of the file to where the complete file reads:

```
[settings]
enableSplunkWebSSL = 1
server.socket_host = localhost
```

After modifying the file, save it and exit the text editor (e.g., `:wq!`) then run:

```
/opt/splunk/bin/splunk set web-port 8443
systemctl restart Splunkd.service
ss -alnt4 | grep 8443
```

If the output of the `ss` command reads something similar to:

```
LISTEN  0          128          127.0.0.1:8443          0.0.0.0:*
```

Then, congratulations, the reconfiguration was a success. By setting the `server.socket_host` directive to `localhost` in the `web.conf` file, we've made it so where the Splunk web interface only binds to the loopback or "localhost" address `127.0.0.1`. The `splunk set web-port 8443` changes the TCP port that the splunk web interface to port `8443/TCP` instead of port `8000/TCP`.

You're probably wondering:

How do I access the web interface if its only available on the loopback interface?

Why did we change the TCP port that splunk web interface is listening on?

We'll be installing Apache web server, and using it as a reverse proxy to accomplish that feat. To make configuring the reverse proxy for our lab network much easier, we needed port `8000/TCP` to be available. Next up, run the following commands:

```
apt-get update
apt-get -y install apache2 apache2-utils
a2enmod proxy proxy_http ssl
touch /etc/apache2/.htpasswd
htpasswd /etc/apache2/.htpasswd [username]
systemctl stop apache2.service
a2dissite 000-default.conf
```

The `apt-get` commands above are responsible for installing the `apache2` web server. Next up, the `a2enmod` command tells `apache2` that we want to enable the `proxy`, `proxy_http`, and `ssl` modules. Next up, we use the `touch` command to create the file `/etc/apache2/.htpasswd`. This file will be used by `apache2` to define what usernames and passwords are allowed to access the Splunk web interface, using the `htpasswd` command to add usernames and hashed passwords to the `.htpasswd` file. For the `htpasswd` command, be sure to replace `[username]` with the username you want to use to authenticate to Splunk. For example, I used the username `ayy`. The `htpasswd` command will then prompt students to enter a password for the user they just created. You can re-use the password you configured for the Splunk administrative user, or create something brand new. Just remember to store the credentials in a password manager for safe keeping. Next up, we use `systemctl` to stop the `apache2` service, and the command `a2dissite` to disable the `000-default.conf` apache site configuration.

Next, students will need modify the file `/etc/apache2/ports.conf`. Ignoring all of the comment lines in the file (e.g., any line that begins with a hashmark/octothorpe `[#]`), here are the default contents of the file:

```
Listen 80
```

```
<IfModule ssl_module>  
    Listen 443  
</IfModule>
```

```
<IfModule mod_gnutls.c>  
    Listen 443  
</IfModule>
```

Using your favorite text editor, remove all of these lines, and replace them with this single line:

```
Listen 8000
```

When finished, save the file and exit the text editor. Next up, we'll be creating a custom site configuration file to proxy any connections to `https://[SIEM IP address]:8000` to `https://127.0.0.1:8443/`. Using your favorite text editor, create the file `/etc/apache2/sites-available/splunk-proxy.conf`:

```
<VirtualHost *:8000>  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
    SSLEngine on  
    SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem  
    SSLCertificateKeyFile  /etc/ssl/private/ssl-cert-snakeoil.key  
    SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1  
    SSLCipherSuite          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-  
SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-  
CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-  
AES256-GCM-SHA384  
    SSLHonorCipherOrder off  
    SSLSessionTickets off  
    SSLProxyEngine On  
    SSLProxyVerify none  
    SSLProxyCheckPeerCN off  
    SSLProxyCheckPeerExpire off  
    ProxyRequests On  
    ProxyPass / https://127.0.0.1:8443/  
    ProxyPassReverse / https://127.0.0.1:8443/  
    <Proxy *>  
        Order deny,allow  
        Allow from all  
        AuthType Basic  
        AuthName "Password Required"  
        AuthUserFile /etc/apache2/.htpasswd  
        Require valid-user  
    </Proxy>  
</VirtualHost>
```

Please be aware, that this site configuration uses a self-signed SSL certificate. Additionally, because the Splunk application also uses a self-signed SSL certificate, there are certain configuration options that are required in order to proxy the connection. As a result of that, some SSL parameters, and certificate settings are not checked. When finished, save the file, exit your text editor, and run the following commands:

```
a2ensite splunk-proxy.conf
systemctl start apache2.service
ss -a|nt | grep 8000
```

These commands are used to enable the `splunk-proxy.conf` site configuration we just made, then start the `apache2` web service. Finally, we use the `ss` and `grep` commands to confirm that the `apache` web server is listening on port `8000/TCP`. The output should look something like this:

```
LISTEN 0          511                *:8000             *:*
```

Now comes the fun part: open your web browser and navigate to `https://[SIEM IP address]:8000`, substituting `[SIEM IP address]` to reflect the IP address assign to your lab's SIEM VM. As usual, your web browser is going to scream bloody murder because you've committed the crime of connecting to a website with a self-signed SSL certificate. As usual, ignore these complaints and ask the web browser to let you proceed anyway. Upon attempting to proceed further, a small pop-up will appear with the text:

http://[SIEM IP address]:8000 is requesting your username and password. The site says: "Password Required"

Followed by two input boxes to enter a username and password. Enter the username and password you configured using the `htpasswd` command earlier, then hit enter, or click *OK* (fun fact: the `Authname` field in `splunk-proxy.conf` file controls what "The site says"). If entered correctly, the Splunk web application should appear. Congratulations, you've successfully placed the Splunk web app behind an SSL reverse proxy, with HTTP basic auth.

```
root@siem:~# apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
root@siem:~# apt-get -y install apache2 apache2-utils haveged
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
root@siem:~# a2enmod proxy proxy_http ssl
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@siem:~# touch /etc/apache2/.htpasswd
root@siem:~# htpasswd /etc/apache2/.htpasswd ayy
New password:
Re-type new password:
Adding password for user ayy
root@siem:~# a2enmod proxy proxy_http ssl
Module proxy already enabled
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
root@siem:~# touch /etc/apache2/.htpasswd
root@siem:~# htpasswd /etc/apache2/.htpasswd ayy
New password:
Re-type new password:
Adding password for user ayy
root@siem:~# systemctl stop apache2.service
root@siem:~# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Continued to *fig. 18-10*

18-9: Begin by downloading `apache2`, `apache2-utils`, and `haveged` via the `apt-get` command. Next run the command `a2enmod` to enable the `proxy`, `proxy_http` and `ssl` modules. Next, create the file `/etc/apache2/.htpasswd`, and use the `htpasswd` command to create a username and password to add to the file. Finally, use `systemctl` to stop the `apache2` service, and `a2dissite` to disable the default `apache2` site configuration.

Continued from *fig. 18-9*

```
root@siem:~# vi /etc/apache2/ports.conf
```

```
# If you just change the port or add more ports here, you will likely also  
# have to change the VirtualHost statement in  
# /etc/apache2/sites-enabled/000-default.conf  
  
Listen 8000
```

```
:wq!
```

```
root@siem:~# vi /etc/apache2/sites-available/splunk-proxy.conf
```

```
<VirtualHost *:8000>  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
    SSLEngine on  
    SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem  
    SSLCertificateKeyFile  /etc/ssl/private/ssl-cert-snakeoil.key  
    SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1  
    SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-EC  
    SA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-C  
    HACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384  
    SSLHonorCipherOrder off  
    SSLSessionTickets off  
    SSLProxyEngine On  
    SSLProxyVerify none  
    SSLProxyCheckPeerCN off  
    SSLProxyCheckPeerExpire off  
    ProxyRequests On  
    ProxyPass / https://127.0.0.1:8443/  
    ProxyPassReverse / https://127.0.0.1:8443/  
    <Proxy *>  
        Order deny,allow  
        Allow from all  
        AuthType Basic  
        AuthName "Password Required"  
        AuthUserFile /etc/apache2/.htpasswd  
        Require valid-user  
    </Proxy>  
</VirtualHost>
```

```
:wq!
```

Continued to *fig. 18-11*

18-10: Next up, students will need to edit the `/etc/apache2/ports.conf` file. Delete every line in the file (that is not commented out) and replace it with the line: `Listen 8000`, then save the file. Afterwards, open your preferred text editor and create the file `/etc/apache2/sites-available/splunk-proxy.conf`, using illustration above and the configuration file on page 1011 for reference.

Continued from *fig. 18-10*

```
root@siem:~# a2ensite splunk-proxy.conf
Enabling site splunk-proxy.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@siem:~# systemctl start apache2.service
root@siem:~# ss -alnt | grep 8000
LISTEN 0          511                *:8000              *:*
```

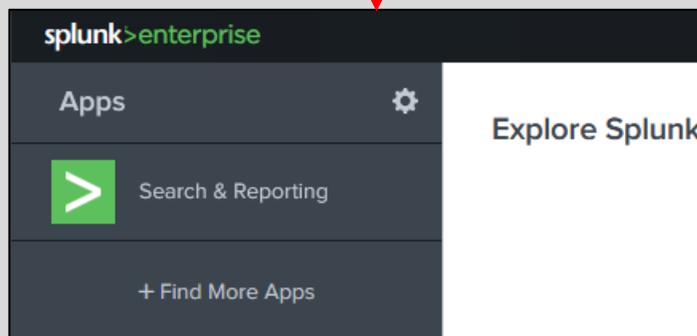
https://172.16.1.3:8000|

Authentication Required - Mozilla Firefox

https://172.16.1.3:8000 is requesting your username and password. The site says: "Password Required"

User Name:

Password:



18-11: with the `ports.conf` file modified and the `splunk-proxy.conf` file created, use the `a2ensite` command to enable the `splunk-proxy.conf` site, then use `systemctl` to start the `apache2` service. Next up, open your favorite web browser and enter `https://[SIEM IP address]:8000` in the URL bar. As usual your web browser will scream bloody murder for daring to connect to a website that uses a self-signed SSL cert, and usual, tell your browser to sod off that you what you're doing, and continue connecting. As soon as you proceed to connect, a small pop-box will appear asking you to enter a username and password. Enter the username and password configured with the `htpasswd` command, and the Splunk web interface should appear. Congratulations, Splunk is now behind an SSL reverse proxy with basic auth configured.

18.2 Installing and Configuring the Universal Forwarder on the IPS VM

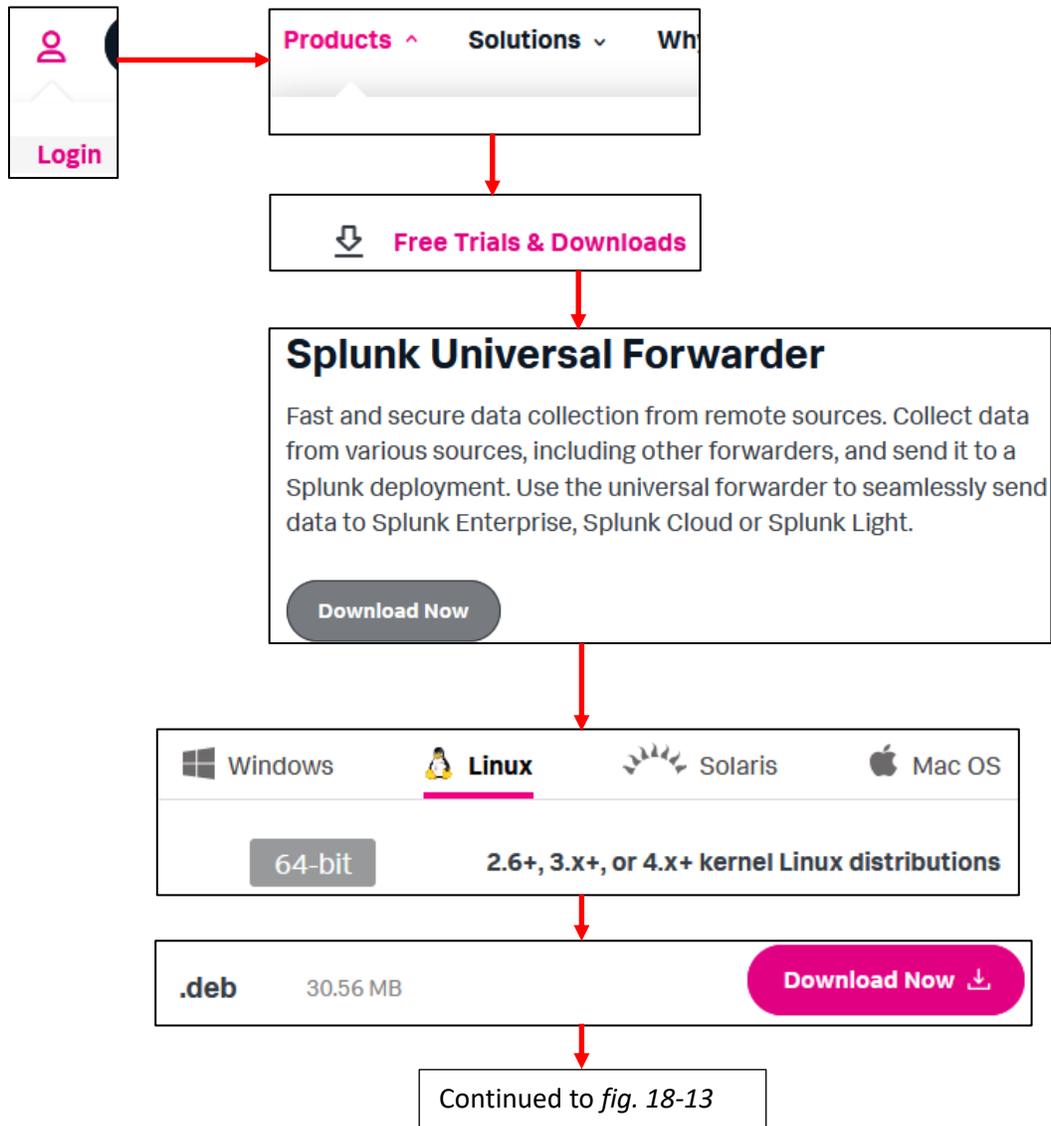
At this point, the SIEM virtual machine is prepared to receive data, and we're ready to install the universal forwarder application on IPS VM. This section will be divided into three major subsections. The first subsection deals with downloading and installing the universal forwarder on to the SIEM VM, and the remaining two subsections guide users on how to configure the universal forwarder to ingest logs for Snort or Suricata. Before we begin, I recommend opening an SSH session to the IPS VM either as the root user, or using `sudo su -` to become root. I also recommend students take a checkpoint/snapshot prior to attempting to install the universal forwarder, just in case.

18.2.1 Downloading and Installing the Universal Forwarder package for the IPS VM

Downloading the universal forwarder is pretty straightforward, and the process is nearly identical to the process we followed to download and install Splunk Enterprise on to the SIEM VM. So, in the interest of saving time (and paper), here is a quick run-down of the tasks to download the installer:

- Log in to splunk.com
- Navigate to *Products > Free Trials & Downloads*
- Scroll down to *Splunk Universal Forwarder* and click the *Download Now* button
- Accept the Splunk license agreement and click the *Start your Download Now* button
- On the next page, under the *Choose Your Installation Package* table, click *Linux*, then click the *.deb Download Now* link in the column that reads *64-bit, 2.6+, 3.x+ or 4.x+ kernel Linux distributions*.
- On the *You're Downloading Splunk Universal Forwarder* page, cancel the browser's download dialogue box, and click on the *Download via Command line (wget)* link
Highlight and copy of the entire `wget` command, then paste it into the SSH session to the IPS VM, to use `wget` to download the `.deb` package
With the package downloaded, run `dpkg -i splunkforwarder-*.deb` to begin the installation process.
- Just like with Splunk Enterprise, the `splunkforwarder` package creates the `splunk` user. Students need to run a series of commands in order to change the owner of the files in `/opt/splunkforwarder`, start the forwarder service, establish persistence for the universal forwarder service, and connect it to the SIEM VM:
 - `chown -R splunk:splunk /opt/splunkforwarder`
 - `/opt/splunkforwarder/bin/splunk start --answer=yes --accept-license`
 - As with the SIEM VM, this command requires students to create an administrative user and password. Be sure to store these credentials in a password manager.
 - `/opt/splunk/bin/splunk stop`
 - `/opt/splunkforwarder/bin/splunk add forward-server [SIEM IP address]:9997`
 - `/opt/splunkforwarder/bin/splunk enable boot-start -systemd-managed 1`

The only new command of any note in the list of commands above is `splunk add forward-server [SIEM IP address]:9997`. Notice how the port number in the command matches the TCP port we used to configure receiving? **This command defines where the logs from the IPS VM will be forwarded to, so it is extremely important that the IP address and port combination is correct.** As usual, students should replace [SIEM IP address] with the IP address of the SIEM VM for their lab environment (e.g., `/opt/splunkforwarder/bin/splunk add forward-server 172.16.1.3:9997`). For students who want to run Snort3, jump to section 18.2.3 Otherwise for Suricata, continue on to section 18.2.2.



18-12: Go back to splunk.com, navigate to *Free Trials & Downloads*, scroll down to the *Splunk Universal Forwarder* entry and select download now. On the next page, select Linux, then click the *.deb* *Download Now* for 64-bit Linux distributions on Kernel versions 2.6+, 3.x+, or 4.x+.

Continued from *fig. 18-12*

I have read, understood and hereby accept the Splunk Software License Agreement.

[↓ Start Your Download Now](#)

USEFUL TOOLS

- Download via [Command Line \(wget\)](#)
- Down

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click [here](#) to select the entire command.

```
platform=linux&
version=8.1.3&
product=universalforwarder&
filename=splunkforwarder-
8.1.3-63079c59e632-linux-
```

Undo
Redo
Cut
Copy

```
root@ips:~# wget -O splunkforwarder-8.1.3-63079c59e632-linux-2.6-amd64.deb
'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86
64&platform=linux&version=8.1.3&product=universalforwarder&filename=splunk
forwarder-8.1.3-63079c59e632-linux-2.6-amd64.deb&wget=true'
root@ips:~# dpkg -i splunkforwarder-*.deb
Selecting previously unselected package splunkforwarder.
(Reading database ... 118480 files and directories currently installed.)
Preparing to unpack splunkforwarder-8.1.3-63079c59e632-linux-2.6-amd64.deb
...
Unpacking splunkforwarder (8.1.3) ...
Setting up splunkforwarder (8.1.3) ...
complete
```

Continued to *fig. 18-14*

18-13: After accepting the Splunk license agreement students will be forwarded to The *You're Downloading Splunk Universal Forwarder* page. Cancel the download dialogue box that appears, and instead locate the link *Download via Command Line (wget)*, copy the link and paste it into the SSH session on the IPS VM to download the .deb package directly to the IPS virtual machine. Afterwards, as the root user, use the dpkg command to install the splunk forwarder package.

Continued from *fig. 18-13*

```
root@ips:~# chown -R splunk:splunk /opt/splunkforwarder/
root@ips:~# /opt/splunkforwarder/bin/splunk start --answer-yes --accept-license

This appears to be your first time running this version of Splunk.

Splunk software must create an administrator account during startup. Otherwise, you cannot log in.
Create credentials for the administrator account.
Characters do not appear on the screen when you type in credentials.

Please enter an administrator username: ayy
Password must contain at least:
  * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
```

```
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done

root@ips:~# /opt/splunkforwarder/bin/splunk stop
Stopping splunkd...
Shutting down. Please wait, as this may take a few minutes.

Stopping splunk helpers...

Done.
root@ips:~# /opt/splunkforwarder/bin/splunk add forward-server 172.16.1.3:9997
Added forwarding to: 172.16.1.3:9997.
root@ips:~# /opt/splunkforwarder/bin/splunk enable boot-start -systemd-managed 1
Important: splunk will start under systemd as user: splunk
Systemd unit file installed at /etc/systemd/system/SplunkForwarder.service.
Configured as systemd managed service.
```

18-14: Use the `chown` command to set the `splunk` user and group as the owner of `/opt/splunkforwarder`, and all of its files. Afterwards, run `/opt/splunkforwarder/bin/splunk start --answer-yes --accept-license` to perform the universal forwarder's first-time start-up tasks, including assigning a username and password to the administrative account. Be sure to save the administrative creds to the universal forwarder to a password manager for later use.

When the initial setup tasks are done, run `/opt/splunkforwarder/bin/splunk stop` to stop the forwarder. Next, run `/opt/splunk/bin/splunk add forward-server [SIEM IP address]:9997`, replacing `[SIEM IP address]` with the IP address of the SIEM VM in the lab environment. Note that the port number, 9997, matches the TCP port used to configure a receiver on the SIEM VM. **This command is responsible for forwarding our logs to the SIEM VM, so make absolutely sure that the IP address and port number are correct.**

Afterwards, run `/opt/splunkforwarder/bin/splunk enable boot-start -systemd-managed 1` to install the `splunk` service, allowing the universal forwarder to start automatically on boot. Do not start the universal forwarder serve back up yet. Students need to install either Snort or Suricata add-ons first. Snort's add-on is covered in section 18.2.3, while Suricata is covered in section 18.2.2

18.2.2 Installing the Suricata TA

In order to process logs from Suricata, we need to install an add-on to the IPS virtual machine. Open your preferred web browser, and in the url bar enter: <https://splunkbase.splunk.com>

If students aren't still logged in to their splunk.com accounts, log in to splunkbase using the splunk.com credentials. The top of the splunkbase page has a searchbar in order to locate apps for Splunk. We need to download the app *TA for Suricata*. Students can search for the app in the searchbar, or use this link to the latest version (as of writing this), 2.3.4:

<https://splunkbase.splunk.com/app/4242/>

On the page titled *TA for Suricata*, click the *Download* button, accept the license agreements (without reading them, as is tradition), then click the *Agree to Download* button to download the .tgz file. When the download completes, students will need to transfer the file to the IPS VM, using either WinSCP (Windows) or the `scp` command (Linux/MacOS). Assuming MacOS/Linux users configured the `~/.ssh/config` file, the command to copy the file over to the IPS VM should be:

```
scp ~/Downloads/ta-for-suricata_234.tgz ips:~/
```

Or, if ssh as the root user was enabled:

```
scp ~/Downloads/ta-for-suricata_234.tgz ipsroot:~/
```

With the file copied over to the IPS VM, it needs to be installed to `/opt/splunkforwarder/etc/apps`. SSH over to the IPS VM and become the root user (if you don't already have a session as the root user opened already). If students copied the .tgz file over to the IPS VM as the root user, run the following command:

```
cp ~/ta-for-suricata_234.tgz /opt/splunkforwarder/etc/apps
```

Otherwise, run:

```
cp /home/[IPS user account name]/ta-for-suricata_234.tgz /opt/splunkforwarder/etc/apps
```

Substitute `[IPS user account name]` with the username students created for the IPS VM during initial setup. Then as the root user, run the following commands:

```
cd /opt/splunkforwarder/etc/apps
tar -xvzf ta-for-suricata_234.tgz
vi TA-suricata-4/default/inputs.conf
```

By default, the file looks like this:

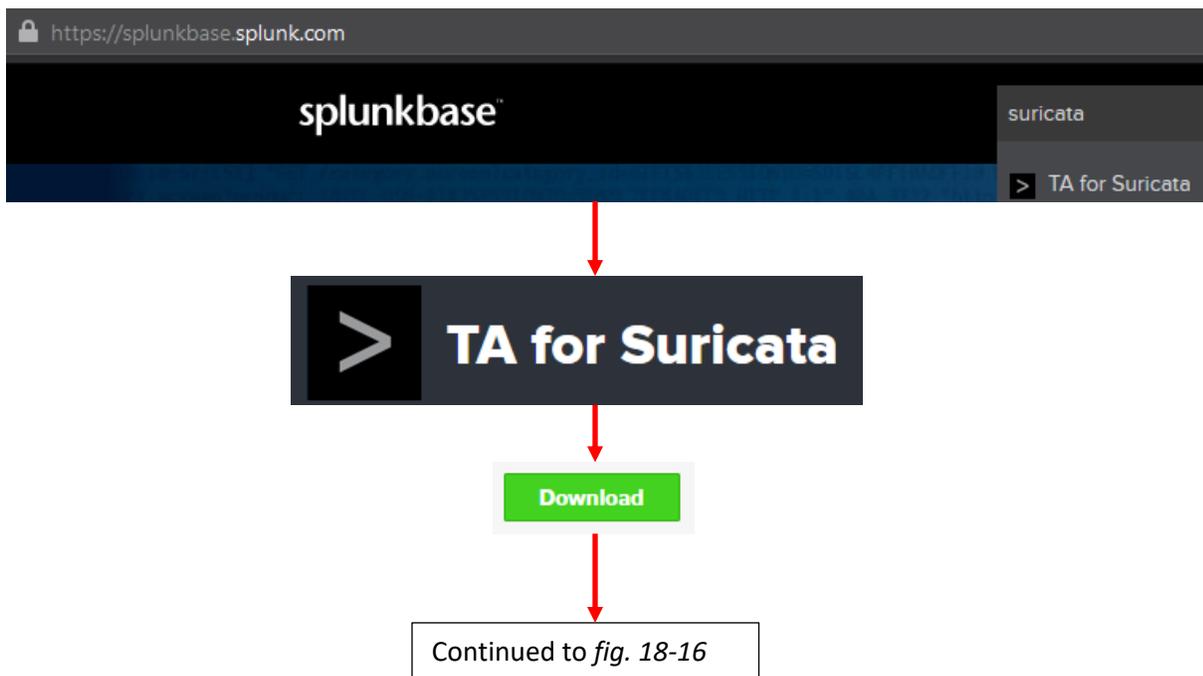
```
[monitor:///var/log/suricata/eve.json]
host = splunk-nat-sec
sourcetype = suricata
index = suricata
```

Using a text editor (e.g., vi), modify the host and index lines to where the file looks like this:

```
[monitor:///var/log/suricata/eve.json]
host = IPS-VM
sourcetype = suricata
index = main
```

Save the edits to the `inputs.conf` file, and exit the text editor. Jump to section 18.3 to continue.

Note: The `main` index is the default index for Splunk. Professionals usually recommend creating a separate index for IDS/IPS events, which is why this file defaults to the `suricata` index. However, since the `suricata` index does not exist on the Splunk Enterprise instance on the SIEM VM, we're just going to stick it on `main`, and creating/configuring a new index can be left as an exercise for you later if you're curious and want to continue your Splunk training.



18-15: Open a web browser and navigate to <https://splunkbase.splunk.com>. Search for the app TA for Suricata, and on its page, click the *Download* button. The current version is 2.3.4

Continued from *fig. 18-15*

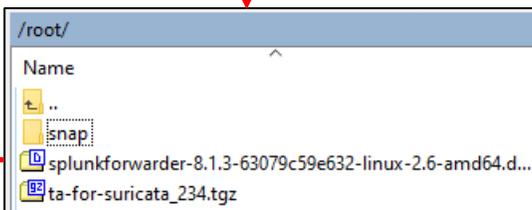
Accept License Agreements

I have read the terms and conditions of this license and agree to be bound by them.

I consent to Splunk sharing my contact information with the publisher of this app so I can receive more information about the app directly from the publisher.

Agree to Download

Downloads - root@IPS - WinSCP



```
trobinson@trobinsons-MacBook-Pro ~ % scp ~/Downloads/ta-for-suricata_234.tgz ipsroot:~/ta-for-suricata_234.tgz
100% 20KB 1.2MB/s 00:00
```

Continued to *fig. 18-17*

18-16: Next, accept the license agreement and download the .tgz file to your Windows/MacOS/Linux workstation. Once downloaded, transfer the file to the IPS VM, using the scp protocol. Windows students can use *WinSCP*, while MacOS/Linux students can use the scp command to accomplish this task.

Continued from *fig. 18-16*

```
root@ips:~# pwd
/root
root@ips:~# ls -al ta-for-suricata_234.tgz
-rw-r--r-- 1 root root 20480 Apr 24 23:46 ta-for-suricata_234.tgz
root@ips:~# cp ~/ta-for-suricata_234.tgz /opt/splunkforwarder/etc/apps/
root@ips:~# cd /opt/splunkforwarder/etc/apps/
root@ips:/opt/splunkforwarder/etc/apps# tar -xzf ta-for-suricata_234.tgz
```

```
root@ips:/opt/splunkforwarder/etc/apps# vi TA-suricata-4/default/inputs.conf
```

```
[monitor:///var/log/suricata/eve.json]
host = IPS-VM
sourcetype = suricata
index = main
```

```
:wq!
```

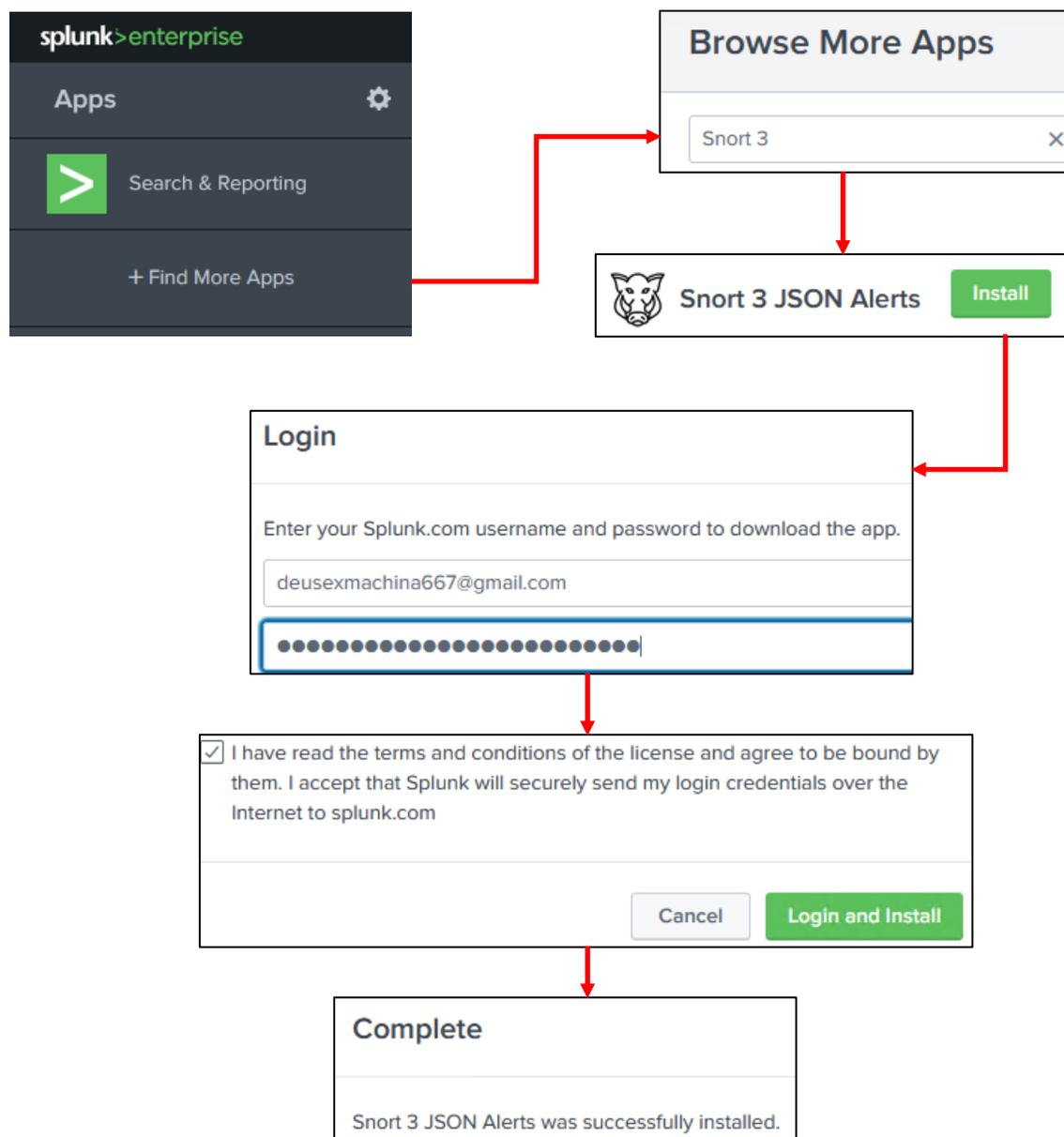
18-17: With the .tgz file copied to the IPS VM, it now needs to be copied to /opt/splunkforwarder/etc/apps, using the cp command. Then as the root user, students will need to use the tar command to decompress the .tgz file, and edit the /opt/splunkforwarder/etc/apps/TA-suricata-4/default/inputs.conf file to match the content in the illustration above. When finished, save the inputs.conf file, and exit the text editor.

18.2.3 Installing the Snort3 JSON Alerts App

Snort3 requires the Splunk app to be installed on both the indexer (the SIEM VM) as well as the log collector (the IPS VM), making it a two-part installation.

18.2.3.1 Installing Snort3 JSON Alerts on the SIEM VM

Open a web browser and log in to the Splunk web interface in the SIEM VM. On the left side of the page, under the *Apps* listing, click on *+ Find More Apps*. On the *Browse More Apps* page, click the search bar on the left side of the page, and search for "Snort 3". Look for the application labeled Snort 3 JSON Alerts, and click the green *Install* button. Students will be prompted to enter their splunk.com username and password, as well as click the license agreement checkbox before clicking the *Login and Install* button. After a moment or two, students should receive a notification confirm that the Snort 3 JSON alerts app has been successfully installed.



18-18: Apparently the Snort 3 JSON Alerts app recommends installing it on both log collectors and indexers in order to make sure the data is normalized correctly, so in addition to installing this app on the IPS VM, we'll be installing it on the SIEM VM as well. Fortunately, the process is dead simple: Log into the Splunk web interface on the SIEM VM, and on the left, under the *Apps* listing, click the *+ Find More Apps* option. On the *Browse More Apps* page, search for Snort 3, locate the *Snort 3 JSON Alerts* app, then click *Install*. Students will be prompted for their splunk.com credentials, and will need to agree to abide by the terms and conditions of the app license before clicking the *Login and Install* button. After a moment or two, a small pop-up window should appear, confirming successful installation of the Snort 3 JSON Alerts app.

18.2.3.2 Installing Snort 3 JSON Alerts on the IPS VM

Note: The current version of the Snort 3 JSON Alerts application on splunkbase was version 1.0.3 as of mine writing this. Of course, software is always subject to change, and that means that the splunkbase URL and/or filename downloaded may change as well.

In order to install the Snort 3 JSON Alerts app on to the IPS VM, students will need to log in to splunkbase.splunk.com, using their splunk.com credentials. On the searchbar on the main page, search for Snort 3 JSON Alerts, or input the following URL:

```
https://splunkbase.splunk.com/app/4633/
```

Click the *Download* button, accept the the license agreements, then click the *Agree to Download* button to download `snort-3-json-alerts_103.tgz`. Next, using the `scp` protocol, students will need to copy the `.tgz` file to the IPS VM. Windows users can do this using WinSCP, while Linux/MacOS users can use the `scp` command. Assuming students configured the `~/.ssh/config` file, and configured SSH as the root user on the IPS VM, the following command can be used:

```
scp ~/Downloads/snort-3-json-alerts_103.tgz ipsroot:~/
```

Otherwise, try:

```
scp ~/Downloads/snort-3-json-alerts_103.tgz ips:~/
```

With the `.tgz` file copied to the IPS VM, SSH to the IPS VM and become the root user. If students copied the `.tgz` file to the IPS VM as the root user try:

```
cp ~/snort-3-json-alerts_103.tgz /opt/splunkforwarder/etc/apps
```

Otherwise, try:

```
cp /home/[IPS user account name]/snort-3-json-alerts_103.tgz /opt/splunkforwarder/etc/apps
```

Substitute `[IPS user account name]` with the username students created for the IPS VM during initial setup. Then as the root user, run the following commands:

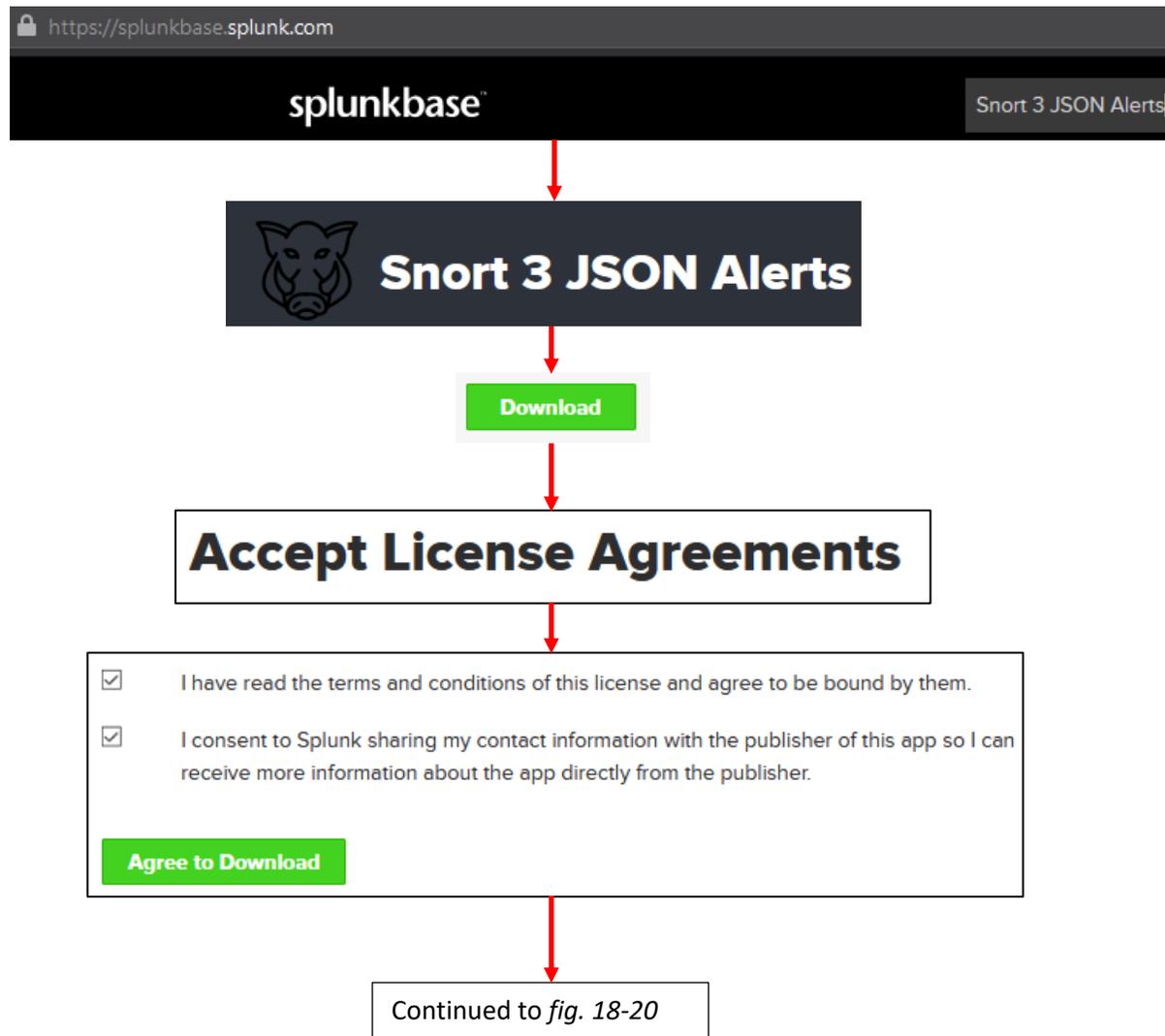
```
cd /opt/splunkforwarder/etc/apps
tar -xzvf snort-3-json-alerts_103.tgz
mkdir TA_Snort3_json/local
vi TA_Snort3_json/local/inputs.conf
```

As usual, if students prefer another text editor to `vi`, that's fine. Create the `/opt/splunkforwarder/etc/apps/TA_Snort3_json/local/inputs.conf` file with the following content:

```
[monitor:///var/log/snort/*alert_json.txt*]  
sourcetype = snort3:alert:json
```

```
[monitor:///var/log/snort/*appid-output.log*]  
sourcetype = snort3:openappid:json
```

When finished, save the file, and exit the text editor. Jump to section 18.3 to continue.



18-19: Unsurprisingly, the process of installing the Snort 3 JSON Alerts app to the IPS VM is similar to the Suricata TA: Login to <https://splunkbase.splunk.com> with your splunk.com credentials, and search for Snort 3 JSON Alerts. On the app page, click the *Download* button, accept the license agreement terms, and click the *Agree to Download* button to download the .tgz file to your workstation.

Continued from *fig. 18-19*

Downloads - root@IPS - WinSCP

```
/root/  
Name  
..  
snap  
snort-3-json-alerts_103.tgz
```

```
trobinson@trobinsons-MacBook-Pro ~ % scp ~/Downloads/snort-3-json-alerts_103.tgz ipsroot:~/  
snort-3-json-alerts_103.tgz 100% 22KB 1.3MB/s 00:00  
trobinson@trobinsons-MacBook-Pro ~ %
```

```
root@ips:~# cp ~/snort-3-json-alerts_103.tgz /opt/splunkforwarder/etc/apps/  
root@ips:~# cd /opt/splunkforwarder/etc/apps/  
root@ips:/opt/splunkforwarder/etc/apps# tar -xzvf snort-3-json-alerts_103.tgz  
TA_Snort3_json/  
TA_Snort3_json/default/  
TA_Snort3_json/default/app.conf  
TA_Snort3_json/default/eventtypes.conf  
TA_Snort3_json/default/tags.conf  
TA_Snort3_json/default/props.conf  
TA_Snort3_json/default/inputs.conf  
TA_Snort3_json/COPYING  
TA_Snort3_json/metadata/  
TA_Snort3_json/metadata/default.meta  
TA_Snort3_json/README  
TA_Snort3_json/static/  
TA_Snort3_json/static/appIcon.png  
TA_Snort3_json/static/appIconAlt.png  
TA_Snort3_json/static/appIcon_2x.png  
TA_Snort3_json/static/appIconAlt_2x.png  
TA_Snort3_json/LICENSE  
root@ips:/opt/splunkforwarder/etc/apps# mkdir TA_Snort3_json/local  
root@ips:/opt/splunkforwarder/etc/apps# vi TA_Snort3_json/local/inputs.conf
```

```
[monitor:///var/log/snort/*alert_json.txt*]  
sourcetype = snort3:alert:json  
  
[monitor:///var/log/snort/*appid-output.log*]  
sourcetype = snort3:openappid:json
```

```
:wq!
```

18-20: Next, students will need to use the scp protocol to copy the .tgz file to the IPS VM. Windows users can use WinSCP, while MacOS/Linux users can use the scp command. With the file on the IPS VM, use the cp command to move it to /opt/splunkforwarder/etc/apps, then use the tar command to decompress it. Finally, use the mkdir command to create /opt/splunkforwarder/etc/apps/TA_Snort3_json/local, then create the file /opt/splunkforwarder/etc/apps/TA_Snort3_json/local/inputs.conf using vi or any other text editor. Use the illustration above to create the file, and when finished, save the changes and exit the text editor.

18.3 Restarting the Splunk Forwarder, and Testing Functionality

SSH to the IPS VM, become the root user, then run the following commands:

```
chown -R splunk:splunk /opt/splunkforwarder
systemctl start SplunkForwarder.service
systemctl status SplunkForwarder.service
ss -ant | grep 9997
```

The `chown` command is ensure that all of the files (including the new Snort/Suricata apps installed) are all owned by the splunk user and group, so there are no permission problems from the service attempting to access its files. Next, we use `systemctl` to start the splunk forwarder, then confirm that it is in the active (running) state. Finally, we use `ss -ant` and `grep`, searching for the string "9997". This command is used to see whether or not the forwarder on the IPS VM able to connect to the SIEM VM on port 9997/TCP. If everything is working correctly, students should see output similar to:

```
ESTAB  0      0          172.16.1.4:42150      172.16.1.3:9997
```

The next step involves triggering IDS events to see if we can get them to show up in Splunk on the SIEM VM. Open up an SSH session to the Kali VM, then run the following commands:

```
curl -I 172.16.2.3
timeout -s KILL 30 wget -m 172.16.2.3
nikto -host 172.16.2.3
```

The `curl` command confirms that Snort or Suricata on the IPS VM is bridging between the IPS1 and the IPS2 network segments. Students should get the output `HTTP/1.1 200 OK`, confirming the IDS software is bridging the network segments correctly. Next, we use the command line utility, `timeout` to run `wget` with the `-m` option to attempt to download (mirror) all the contents of 172.16.2.3. The `timeout` command is configured to kill the `wget` process after 30 seconds. Finally, `nikto` is a very aggressive web application scanner that we will be using to probe the Metasploitable 2 VM in order to generate attack traffic.

Note: When running `nikto`, if you get the message: `+ 0 hosts tested`, try this instead:

```
nikto -host http://172.16.2.3
nikto -host http://172.16.2.3:80
```

Alternatively, if you have other preferred network scanning tools (like `nmap`), try them instead:

```
nmap -Pn -vv -A 172.16.2.3
```

Our only goal here is to generate attack traffic for the IPS VM to log, so we can review it in Splunk.

After running these commands open a web browser, and log into the Splunk web interface on the SIEM VM. On the login page, select the *Search & Reporting* app to bring up the search interface. Over to the right of the search bar is the text *Last 24 hours*. I recommend clicking on this, and setting it to Last 15 minutes. Next, we need some queries to run.

If students are using Suricata, try:

```
sourcetype=suricata event_type=alert | stats count by alert.signature
```

The screenshot shows a Splunk search interface. At the top, the search query is: `1 sourcetype=suricata event_type=alert` and `2 | stats count by alert.signature`. Below the query, it indicates **184 events** for the time range `(4/24/21 8:00:00.000 PM to 4/25/21 8:20:52.000 PM)`. The interface has tabs for **Events**, **Patterns**, **Statistics (13)**, and **Visualization**. Under the **Statistics** tab, there are options for `20 Per Page`, `Format`, and `Preview`. The main content area shows a table with the following data:

alert.signature
ET EXPLOIT D-Link DSL-2750B - OS Command Injection
ET INFO Request to Hidden Environment File
ET POLICY Http Client Body contains pwd= in cleartext
ET POLICY Possible Kali Linux hostname in DHCP Request Packet
ET WEB_SERVER ColdFusion administrator access
ET WEB_SERVER ColdFusion componentutils access
ET WEB_SERVER Possible CVE-2014-6271 Attempt
ET WEB_SERVER Possible CVE-2014-6271 Attempt in Headers
SURICATA Applayer Detect protocol only one direction

```
sourcetype=suricata event_type=http | stats count by http.http_user_agent, http.url
```

The screenshot shows a Splunk search interface. At the top, the search query is: `1 sourcetype=suricata event_type=http` and `2 | stats count by http.http_user_agent, http.url`. Below the query, it indicates **8,874 events** for the time range `(4/24/21 8:00:00.000 PM to 4/25/21 8:24:56.000 PM)` with `No Event Sampling`. The interface has tabs for **Events**, **Patterns**, **Statistics (8,563)**, and **Visualization**. Under the **Statistics** tab, there are options for `20 Per Page`, `Format`, and `Preview`. The main content area shows a table with the following data:

http.http_user_agent	http.url
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/FormMail-clone.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/admin.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/administrator.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/authLogin.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/banner.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/bb-hist.sh
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/book.cgi
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/cgi-bin/
<code>() { : ; }; echo 93e4r0-CVE-2014-6271: true;echo;echo;</code>	/cgi-bin/FormMail-clone.cgi

These queries utilize the suricata eve.json, looking for IDS alerts (event_type=alert) and doing a statistical sort based on the signature name, while the second query utilizes suricata's HTTP logs (event_type=http), sorting the HTTP traffic logged by user-agent and URL requested.

These queries utilize both the Snort3 JSON alerts, as well as the openappid JSON sourcetypes. The first query looks at snort rule alerts and statistically sorts them by the contents of the rule message field, while the second query utilizes the openappid listener, querying the openappid JSON log, statistically sorting events by the http URL and user-agent.

Please bear in mind that the method used to generate attack traffic and the sample queries above are only meant to serve as a proof of concept demonstration. **The only purpose this exercise is supposed to serve is to confirm that traffic is being logged by the IPS virtual machine, then being forwarded to the SIEM VM.** This exercise isn't meant to be a direct comparison of the efficacy of one IDS software platform over another, or meant to prove that one IDS platform is better than another.

If students were able to query the Snort or Suricata logs on the SIEM VM, then congratulations! The baseline lab environment is complete! It is recommended for students to take a snapshot/checkpoint of both the IPS as well as the SIEM virtual machines in this known good condition, and fully configured. If students are running into problems however, take a look at section 18.4 for some troubleshooting recommendations

Note: There are tons of other ways you can test out the IDS/IPS software on the IPS VM. Nikto just happened to be the path of least resistance. Don't be afraid to experiment on your own.

18.4 Troubleshooting Recommendations

If students are experiencing problems getting data into splunk, even after running "the test battery" above, bear in mind there are a lot of moving parts that all need to be working in perfect harmony in order to produce results from the queries. As always, its important to start with the basics and work from there

Is Snort/Suricata running? In Chapter 17, section 17.4 (pp. 992-993), we covered a lot of troubleshooting steps for Snort and Suricata, but below are some quick recommendations:

Consider confirming that either Snort or Suricata is running by using systemctl to confirm the status of the service:

```
systemctl status snort3.service
systemctl status suricata.service
```

Check and confirm that the kali VM can reach the metasploitable 2 VM. By default, the kali VM has the IP address 172.16.2.2 and is located on the IPS1 network segment/virtual switch, while Metasploitable 2 has the IP address 172.16.2.3 and is located on the IPS2 segment. Of course, if students are using a different IP address scheme, adjust the IP addresses as necessary, but the important thing to note is that both devices need to have IP addresses, and need to be on their respective network segments, traversing over the bridge that the Snort or Suricata service provide. Make sure to check that the Metasploitable 2 VM has an IP address by logging in to its console and running the ifconfig command. Use the command `curl 172.16.2.3` (substituting

the IP address as necessary) and verify that the metasploitable 2 banner is returned. At the very least, this can be used to confirm that the kali VM can reach the metasploitable 2 VM and generate traffic.

Try intentionally powering off the IPS VM, or running the command `systemctl stop snort3.service/suricata.service`, then running the `curl` command mentioned above. The goal of this exercise is to intentionally introduce a failure condition to verify that the network traffic is traversing the network bridge the IPS VM is providing. If students are somehow able to still connect to the metasploitable2 VM from the kali VM, there is an alternate path that the network traffic is using to connect the two virtual machines together, and this should not be happening. Check the configuration settings of the Metasploitable 2 VM and the Kali VM to ensure that they are connected to the correct network segments, and that the correct network interfaces on the IPS VM are connected and bridging the correct network segments as well.

If students are running snort, run the command `ls -al /var/log/snort`. There should be at least two files in this directory: `alert_json.txt` and `appid-output.log`. Both of these files should be above zero bytes. If the filesize is greater than zero, that indicates that they are logging *some sort of data*.

If students are running suricata, run `ls -al /var/log/suricata` instead. The only file we care about here is the file `eve.json` and it being a size greater than zero bytes.

If all else fails, `/var/log/syslog` may be logging some sort of a failure condition related to snort or suricata. Try running:

```
grep suricata /var/log/syslog
```

```
grep snort /var/log/syslog
```

Is the Splunk Enterprise instance on the SIEM VM running? The quickest way to determine this is pointing a web browser at `https://[SIEM IP address]:8000` and attempting to log in to the Splunk web interface.

Is there a receiver configured on the Splunk Enterprise instance on the SIEM VM? Take a look at [section 18.1.3.2](#) (pp. 1005-1006), where we covered setting up a receiver to accept logs from the IPS VM. When done, be sure to run the `ss -alnt4 |grep 9997` command to confirm that there is a TCP network socket listening for connections on the SIEM VM.

Is the disk full on the SIEM VM? Run the command `df -h` and monitor for available disk space on the SIEM VM. If the disk is full, then students may want to consider acquiring a larger disk, expanding the disk space on the SIEM VM, finding a way to free up space, etc.

Has the license quota been exceeded for the day? Remember that Splunk free will only index up to 500MB of logs per day, unless students were able to acquire a developer license (or actually purchased a standard license). Students may check their license quota by navigating to *Settings > Licensing* on the Splunk web interface on the SIEM VM. Remember that if the idea of a license

quota for log management doesn't set well, then I provided alternatives to Splunk that students may attempt to implement instead.

Is the Universal Forwarder operating correctly? There are a lot more things that need to be checked for the Universal Forwarder on the IPS VM, than there are on the SIEM VM. Let's start with the basics: Run `systemctl status SplunkForwarder.service` and verify the service is running. Next, run `ss -aIn | grep 9997`. Students are looking for results that start with ESTAB to show that the IPS VM has established a connection to the receiver on the SIEM VM on port 9997/TCP. Remember that all of the files in `/opt/splunkforwarder` should be owned by the `splunk` user and group. If students get any errors related to file permissions, try running `chown -R splunk:splunk /opt/splunkforwarder` to attempt to resolve those issue.

Are the Snort 3 JSON Alerts/Suricata TA apps configured correctly? Ensure that either the suricata TA or Snort 3 JSON alerts app are installed in the `/opt/splunkforwarder/etc/apps` directory, and are owned by the `splunk` user and group. For the Suricata TA, ensure that the `/opt/splunkforwarder/etc/apps/TA-Suricata-4/default/inputs.conf` file is configured to log to the "main" index. We covered this in section 18.2.2.

For the Snort 3 JSON alerts app, students should have created the file

```
/opt/splunkforwarder/etc/apps/TA_Snort3_json/local/inputs.conf
```

pointing to `/var/log/snort/alert_json.txt`, and `/var/log/snort/appid-output.log`. We covered this in [section 18.2.3.2](#) (pp. 1026-1028). Additionally, remember that they recommend installing the Snort 3 JSON alerts app on the indexer (the SIEM VM) as well.

When in doubt, check splunkd.log. On the SIEM VM, try checking:

```
/opt/splunk/var/log/splunk/splunkd.log
```

and on the IPS VM, try checking:

```
/opt/splunkforwarder/var/log/splunk/splunkd.log
```

I can't tell students what exactly they should be looking for, but reading the logs may help to uncover problems.

18.5 Chapter Review

In this chapter, we covered installing Splunk Enterprise on the SIEM VM, as well as a universal forwarder on the IPS VM. We also covered setting up a receiver on the SIEM VM, and forwarding either Snort 3 or Suricata logs using either the Suricata TA or the Snort 3 JSON alerts app on the IPS VM. Students also experienced generated network traffic and alerts in their lab environment on the Kali VM, using it to probe and attack the Metasploitable 2 VM in order to generate traffic, and thus alerts and logs for us to review on the Splunk web interface on the SIEM VM.

At this point, students have completed the baseline lab environment. Congratulations! Are you looking for some ideas in how you can extend or specialize your lab environment? Maybe some alternative configuration options or lab redesigns suited towards a specific goal? Continue on to Chapter 19, starting on page 1037 to review everything we've covered, and may be get some ideas on ways you can customize your lab to better suit your needs.

I also have a bonus chapter I threw together, with content that I thought would be useful, but didn't necessarily fit into a single chapter. Stuff like automating updates for the Linux virtual machines, and hardening hypervisor hosts in order to maintain network boundaries between the host and the lab environment. Students can check out chapter 20, starting on page 1055 if they're interested.

Chapter 19 Patch Notes

- Welcome to the "Final" chapter of the book
- Technically not the final chapter per se, since the bonus content chapter comes after this but... at this point, all of the core content has been covered
- This one chapter replaces chapters 21,22, and 23 from the first book and just condenses all of that into one chapter.
- Just like in the first book, I provided students with a few ideas on how they can retrofit their lab towards blue team pursuits, red team pursuits or ops/sysadmin pursuits.
- In addition to the network diagrams, and a general explanation of what I was thinking when I made them, students are also provided with links to a variety of tools and resources that other people have made, released for free, and generally consented to me including in this work.
- With thanks to: Mark Russinovich and the Sysinternals team, Tom Kopchak, Florian Roth, Jacob Baines, Netspooky, FireEye, the SANS institute, Michael Sikorski, Andrew Honig, Amanda "Malware Unicorn" Rousseau, Brad Duncan, vulnhub, HackTheBox, TryHackMe, and OverTheWire
- Because I considered systems administration and ops to have a massive subject area, I didn't have much in the way of direct recommendations, which is why I wrote the sidebar discussion "*Obtaining the Guidance you Seek*".
- I also wanted to offer one more final thank you Chris Long for his amazing work that went in to the DetectionLab project.

Chapter 19: End of the Beginning

If you made it this far, congratulations. By now, students should have a fully functional baseline environment. I have nothing left to teach, and you should have most of the knowledge necessary to re-fit your lab environment to better suit your needs. Before we talk about expanding or re-arranging the lab environment, Let's talk about some of the things we've covered together in this book.

19.1 Chapter Review

In chapter 1, students learned why having building a virtual lab environment is a good idea. In general, it's never a bad plan to have a test environment to learn new concepts, or try out new things. That way, accidents are better contained, and experimenting with new technologies or new concepts are much less likely to cause problems for others. In this chapter, students were also provided with a hefty list of prerequisites and recommended software to have installed for Windows, MacOS or Linux.

In chapter 2, I provided students with a list of recommended prerequisite knowledge in order to progress their way through this book. I noted that these were not hard prerequisites, but having the recommended knowledge would make many of the configuration tasks in the coming chapters much easier. I also provided links to a variety of learning resources for students looking to fill in the gaps.

In chapters 3 and 4, students learned about virtualization, and difference between hosted and bare-metal hypervisors. They then discovered how bare-metal and hosted hypervisors handle network connectivity for virtual machines – specialized network segments (e.g., bridged, NAT, host-only, internal-only, etc.) for most hosted hypervisors vs. virtual switches for most bare-metal hypervisors.

In chapter 5, students learned more computer hardware, system resources and their impact to hypervisor performance, as well as specific CPU and system features required to run most modern hypervisors.

In chapter 6, students were introduced to the design of the baseline lab environment. They learned what virtual machines would be used, the virtual network design, resource allocations, and the functionality each VM would provide. Additionally, this chapter provided a couple of recommendations on places in which lab network hardware could be acquired.

Chapter 7 introduced students to password managers. While password management isn't technically a core component of our virtual lab, credential management is extremely important for maintaining secure access to various portions of the lab environment students created.

Chapter 8 served as a jumping off point for the rest of the book, in which students made a decision in what chapters they will be reading going forward, depending on the hypervisor they

wanted to use to host their lab environment. Students were informed that the next 5 chapters were dedicated toward guiding students on how to create the lab environment on various hypervisors – Microsoft Client Hyper-V, Oracle VirtualBox, VMware Workstation, VMware Fusion, and VMware ESXi.

Chapters 9 through 13 are collectively known as the "Hypervisor Setup Chapters". While the specific steps required to set up the lab environment and the terminology for each hypervisor differ, the overall structure of these chapters is nearly identical: Students learn how to install various hypervisors, set up the lab network segments or virtual switches, create their virtual machines, then install the operating systems to the virtual machines that compose the baseline lab environment described in Chapter 6. After creating the first virtual machine, pfSense, students were then redirected to chapter 14 in order to fully configure the pfSense virtual machine in order to provide network services necessary to finish setting up the other virtual machines that comprise the baseline lab environment.

As mentioned above, Chapter 14 was dedicated towards guiding students on how to configure the pfSense virtual machine. Student learned how to configure the pfSense firewall distribution in order to provide a variety of core network services for the lab environment – DNS, HTTP proxy, DHCP, and NTP. Additionally, students also learned how to configure a secure, robust firewall rule policy for the WAN (Bridged), LAN (Management) and OPT1 (IPS1/IPS2) networks. Students learned that these rules serve as the primary method for enforcing segmentation between the virtual network segments, as well as between the virtual lab, and physical networks students have their lab environment connected to.

Chapters 15 and 16 were the routing and remote access chapters for hosted and bare-metal hypervisors, respectively. In these chapters, students learned how to configure static routes on their hosted hypervisor system and/or bastion host, and also learned how to configure SSH connection profiles to remotely access the Linux virtual machines using the SSH protocol. Students running Windows systems were instruction on how to use mRemoteNG to enable SSH access, while Linux and MacOS users were instructed on how to use the `ssh` command. Bare-metal hypervisor students were introduced to the concept of a bastion host, and were provided with options to configure a physical bastion host, or a virtual machine bastion host in order to provide secure access to their lab environment. Students were also provided with optional lessons for configuring key-based authentication, enabling access as the root user over SSH, and disabling password authentication over SSH, instead relying on key-based authentication.

Chapter 17 provided students with a choice between installing Snort 3 or Suricata on the IPS virtual machine in order to utilize the software as a network bridge to connect the IPS1 and IPS2 networks together, while simultaneously being able to log IDS alerts and connection details.

Chapter 18 instructed students on how to install Splunk Enterprise on the SIEM VM, enabling https access to the Splunk web interface, configuring Splunk's free licensing option, and optionally, setting up the Splunk web interface behind an SSL reverse proxy, configured for basic

authentication, in order to mitigate the lack of authentication available in Splunk's free edition. Afterwards students were also guided on how to configure a Splunk universal forwarder on the IPS VM, and configure it to either forward Snort 3 or Suricata IDS logs for review on the SIEM VM. As a part of this setup, students generated a barrage of attack traffic, and were provided a handful of commands to query the logged data in order to confirm that both the Splunk web interface, universal forwarder, and either the Snort or Suricata IDS software were all working correctly.

19.2 Remodeling and Expansion

Some may wonder why I'm calling this chapter the End of the Beginning. Well, it's because this entire book was all about establishing foundations. You learned how to create and maintain a **baseline** lab environment. All I've done is instruct you on how to set up the base components, and even then, there is a ton of customization students can do if they don't like the software or operating systems used in the baseline lab environment:

Don't like pfSense? Maybe consider OPNsense, IPFire, or any other number of firewall distributions out there.

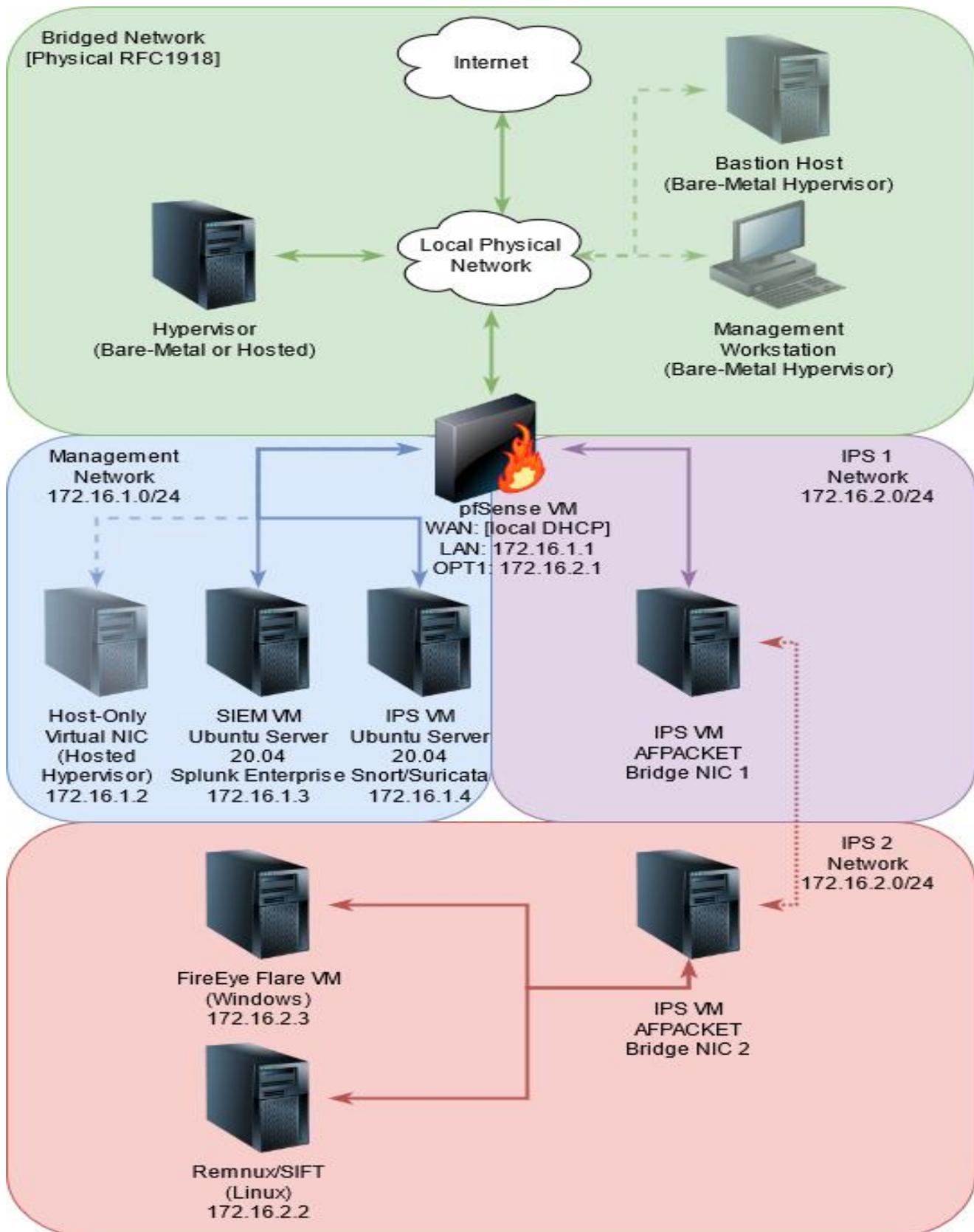
Don't like using Ubuntu Server? Maybe try out Redhat and its variants. Maybe use Gentoo or Arch Linux (if you like pain). Don't care for Linux at all? Considering trying out BSD instead.

Don't like Splunk? Replace it with a log management SIEM of your choice – Wazuh, AlienVault OSSIM, ELK, Security Onion, SIEMonster, MozDef, etc.

Don't care for Snort or Suricata, but still like the idea of a fail-close bridge? Look into using `bridge-utils` or alternate software packages to configure network bridging.

These are just some of the ways in which students may be able to customize the baseline lab environment. What about expanding it?

19.2.3 Outfitting a Malware Analysis Lab



The network diagram above depicts retrofitting the baseline lab environment into a malware analysis lab. Notice that the Kali and Metasploitable 2 virtual machines are gone, replaced with a pair of VMs in the IPS 2 network, marked FireEye FLARE and REMNUX/SIFT.

FireEye's flare VM toolkit is an "overlay" installer package that can be added to any Windows virtual machine running Windows 7 or later. This overlay contains a wide variety of valuable malware analysis tools. For more information, including a run-down of the tools included, requirements, and instructions for installing the overlay, visit:

<https://github.com/fireeye/flare-vm>

Taking Windows Malware Analysis Further

While the FLARE VM overlay has a ton of useful tools, and makes it really easy to install new tools (that are at least available via the chocolatey repository), it isn't the be-all end-all. For example, for one reason or another, the sysinternals suite is not installed by default, and that is absolutely something I would want for dynamic malware analysis, if for no other reason than the sysinternals suite includes Sysmon.

Consider installing Sysmon, and reading up on how to integrate Sysmon logs into Splunk. Remember that you'll have to modify the firewall rules on the pfSense VM to allow the SIEM VM to receive logs from the OPT1/IPS network. I did some blog posts a little while back on how powerful process execution logs (Windows Audit Log Event ID 4688 and/or Sysmon Event ID 1) can be for investigations. Consider having a look here:

<https://hurricanelabs.com/splunk-tutorials/threat-hunting-with-splunk-part-1-intro-to-process-creation-logs/>

Disclaimer: Hurricane Labs is my employer. No, I'm not trying to sell you anything.

In addition to enabling process execution logs, maybe consider enabling powershell transcript logs as well. Here are some details on that, written by one of my colleagues, Tom Kopchak:

<https://hurricanelabs.com/splunk-tutorials/how-to-use-powershell-transcription-logs-in-splunk/>

Even if you decided to remove Splunk and replace it with another SIEM as a part of your lab redesign, Sysmon logs and/or powershell transcript logs are extremely powerful and will help out a lot with dynamic malware analysis. As always, snapshot management is important, but doubly so for malware analysis VMs. Make sure you have a known good snapshot to revert to after doing any dirty work in your windows VM. If students need a trial copy of Windows 10 or Windows Server, check out the sidebar discussion, *Taking your Offensive Security Lab Further*.

This is just the tip of the iceberg. There is a lot more you can do for Windows malware analysis, but this is as good a start as any. Good luck!

The other new virtual machine in the IPS2 network is titled SIFT/REMNUX. SIFT and Remnux are two virtual machine overlays created by various individuals in the information security community, provided by the SANS institute. SIFT is more focused on data forensics and incident response tools, while Remnux is more focused on malware analysis and reverse engineering.

To learn more about SIFT, visit: <https://github.com/teamdfir/sift>

To learn more about Remnux, visit: <https://remnux.org/#distro>

Students can opt to install these virtual machine overlays on individual virtual machines, or both of them combined on a single virtual machine.

Taking Linux Malware Analysis One Step Further

Admittedly, my knowledge of Linux/Unix Malware analysis isn't quite as great as it is with Windows, but in terms of options for logging malicious activity, `auditd` is a pretty good choice. Security Researcher Florian Roth put together a collection of `auditd` rules that can be used to detect malicious activity on a host:

<https://github.com/Neo23x0/auditd>

Alternatively, OSQuery is an option for being able to pull information from Linux, Windows and/or MacOS hosts: <https://osquery.io/>

Here are a couple of other tools that I've had recommended to me for Linux malware analysis:

Elf Parser: <http://elfparser.com/download.html>

Written by Jacob Baines, Elf Parser can read the structure of the ELF binaries and provides a score based on how "suspicious" the file is. This tool can also provide a basic readout on what functions the binary has

Inhale: <https://github.com/netspooky/inhale>

Described as the malware inhaler, this tool (written by Twitter user, @netspooky) can perform initial analysis and classification on various malware samples. This classification data can then be added to a database (Elasticsearch) or dumped to an HTML report.

As always, there are tons of directions you can take your research and lab design. Don't be afraid to experiment. Good luck!

In terms of additional resources, if students plan on setting up individual SIFT and Remnux virtual machines, I would recommend two Ubuntu Linux 20.04 virtual machines with 4GB of RAM and 80GB of disk space each. If students plan on installing the SIFT and Remnux overlays on to a single virtual machine, That VM should have at least 120GB of disk space, and 4GB of RAM allocated.

As for the FLARE VM, I would recommend 4GB of RAM, and 80GB of disk space available for it as well. If students removed the Kali and Metasploitable 2 virtual machines, that would bring the total resource allocations to anywhere from 16.5GB to 20.5GB of RAM, and anywhere from 330GB to 410GB of disk space. Be aware that these totals do not account for snapshots, storing operating system ISOs, or operating system overhead for hosted hypervisors. Realistically, I would say to have at least 800GB of disk space available, and at least 24GB of RAM available. Students may consider powering off virtual machines they are not using in order to save on RAM usage.

Additional Malware Analysis Training and Resources

When it comes to malware analysis in general, there are tons of resources out there on the internet. Here is a brief list of free (or at least affordable) resources to level up your malware analysis experience:

Practical Malware Analysis: This book is published by No Starch Press, written by Michael Sikorski and Andrew Honig. While the book is a little bit long in the tooth, (having been published in 2012), its still highly recommended reading, and a wonderful introduction into the subject matter.

RE101/RE102: This training, written and designed by Amanda "Malware Unicorn" Rousseau, is an introduction to malware reverse engineering, a very complex topic. This training is extremely good quality and is 100% free. Amanda's work is available at:

<https://malwareunicorn.org/#/workshops>

Malware Traffic Analysis Exercises: This is a collection of exercises provided by Brad Duncan. In most of the exercises, students are supplied a packet capture along with other artifacts with the goal of performing an incident response investigation. Students are then expected to write a report about the malware they observed, including an executive summary, details on the malware, the systems it infected, the users running on those systems, and finally an index containing indicators of compromise (e.g., file hashes, IP addresses, domains, URLs, filenames, etc.). This is a great introduction to network intrusion detection, file carving, security operations, and incident response in general. Check it out at:

<https://www.malware-traffic-analysis.net/training-exercises.html>

SOC it to Me

In addition to malware analysis tools and techniques, it may be beneficial to install additional open-source tools utilized by security analysts and engineers. Here are a few recommendations to get you started:

MISP: The MISP project is described as an open-source threat intelligence platform. More than that, it's a threat intelligence platform that is well maintained, features good documentation, an expansive API, and a very active community. More information can be found at:

<https://www.misp-project.org/>

TheHive/Cortex: TheHive is described as a 4-in1 security incident response platform, While Cortex is described as a powerful observable analysis and active response engine. In a nutshell, TheHive is a platform in which security and incident response events can be logged, and tasks associated to the event can be defined and assigned for completion, while Cortex is essentially a platform that can be used to enrich (gather more information about) indicators observed during an investigation. While these two projects are independent, they are usually deployed together. Find out more at:

<http://thehive-project.org/>

GRR Rapid Response: GRR is a project by google. It's described as an incident response framework focused on remote live forensics. In a nutshell, GRR allows analysts to perform a variety of incident response tasks against systems in which the GRR agent is installed. Analysts can do things like request live memory analysis (memory forensics) or request raw file access (disk forensics) to retrieve suspected malware artifacts, etc. More information can be found at:

<https://grr-doc.readthedocs.io/en/latest/>

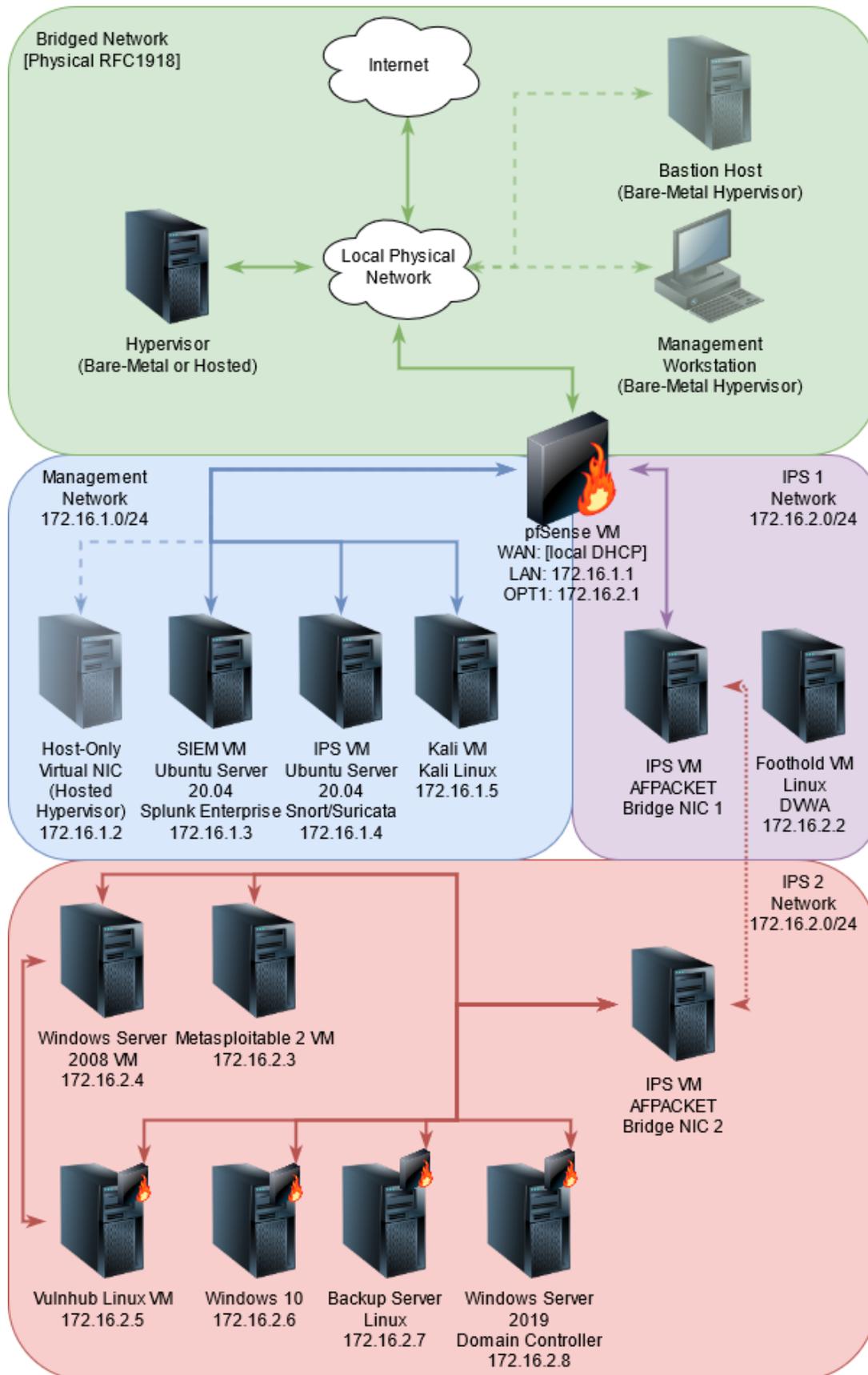
n8n: n8n is described as a workflow automation tool. This tool allows users to define various trigger events and define actions or "nodes" that can be ran in response to those triggers. For example, hypothetically if a new TheHive assignment has been created, an n8n workflow can define that as a trigger. That trigger can then kick off a variety of nodes or responses. For example, a slack message to a particular server and channel to notify users in the channel that a new assignment is available. Learn more at:

<https://docs.n8n.io/#what-is-n8n>

Zeek: Zeek is described as a network security monitoring platform. Where Snort and Suricata are primarily concerned with matching network patterns in signatures and generating alerts (sometimes with little to no context), Zeek passively observes network traffic, and logs it. These logs can then be used to build context around incidents. Zeek is not a replacement for Snort, Suricata or any other IDS/IPS platform, but a companion. Wanna know more? Check out:

<https://docs.zeek.org/en/master/>

19.2.4 Outfitting an Offensive Security/Penetration testing lab



Considering the fact that the baseline lab environment already has a Kali Linux VM, and a Metasploitable 2 VM – one of the oldest known virtual machines still around to teach offensive security concepts, coupled with one of the most commonly used penetration testing distributions – outfitting a lab environment better suited towards offensive security and penetration testing pursuits isn't a far leap. Students can make their testing environment as complex or as simple as they'd like – the only bottleneck being resources available, relative to the number of virtual machines they would like. The network diagram above is just an example of that.

The goal of the lab environment depicted above would be to simulate something relatively close to a real network. For that reason, I elected to move the Kali VM to the management network, behind a separate segment of the pfSense firewall. This would make the Kali VM subject to firewall rules in order to reach the vulnerable virtual machines in the IPS1 and IPS2 networks. This could be used to teach about limiting attack surface, the importance of inbound and outbound firewalls, pivoting, placement of IDS/IPS sensors, etc.

For example, if students wanted to experiment with pivoting, they could configure a VM in the IPS 1 network segment as an initial foothold, only allowing HTTP/HTTPS inbound from the Kali VM in the management work. This foothold VM could then be outfitted with vulnerable web applications such as the Damn Vulnerable Web Application suite:

<https://github.com/digininja/DVWA>

This would require students to exploit the web application and gain code execution before being able to target the virtual machines in the IPS2 network segment.

From there, The IPS2 network could be modified to host any virtual machines the student wishes to experiment with. Microsoft makes it possible to download free trials of various operating system ISOs, while intentionally vulnerable Linux virtual machines can be acquired from vulnhub.com – called "boot 2 root" virtual machines.

The idea would be to create a sprawling "enterprise" network in which gaining deeper access into the network would require pivoting from one host to another. Notice half of the hosts with a small firewall icon on them? The idea would be to allow inbound access to these systems from specific other hosts in the IPS2 network only, meaning that exploiting those VMs would require pivoting from a virtual machine the student has already compromised.

Assuming all of the virtual machines added to the IPS1 and IPS2 networks were allocated 40GB of disk space, and 1GB of RAM for Linux, 4GB of RAM for Windows, students would require, at a minimum, 240GB of additional disk space, and another 15GB of ram to sustain all of these virtual machines at once – on top of the 13GB of RAM and 255GB of disk required for the baseline lab environment, bringing your totals up to 24GB of RAM, and 535GB of disk space – not including snapshots, or operating system overhead for hosted hypervisors. I would say realistically to have a system with at least 32GB of RAM and at least 1TB of available disk space to pull this off reliably.

The other alternative would be to work on a smaller number of virtual machines at a time. The lab network may not be quite as large, but it can be used just as effectively to learn the basics of offensive security and penetration testing.

Taking your Offensive Security Lab Further

If you're an absolute beginner to penetration testing and offensive security, I recommend trying out Offensive Security's Metasploit Unleashed course to get your bearings. Their course material is located at: <https://www.offensive-security.com/metasploit-unleashed/>

The course is 100% free and online. Not to mention that you already have 2/3rds of the virtual machines required for the course (You already got Kali and Metasploitable 2, but they also recommend a Windows VM of some sort), so this training is as good a place to start as any.

As I mentioned above, one of the greatest resources for acquiring intentionally vulnerable virtual machines is the vulnhub project: <https://www.vulnhub.com/>

There are a huge variety of different virtual machines to try out, all teaching slightly different lessons. Some of the virtual machines may feature write-ups while some may not. The best advice I can offer with vulnhub virtual machines is to experiment, persist, and when in doubt, trying using your favorite search engine to see if there is a write-up for the virtual machine you downloaded, if you happen to get stuck.

You'll notice in the diagram for the offensive security lab that I didn't remove the SIEM VM. That's because Splunk (or whatever alternative SIEM you seek to install) alongside security monitoring tools can be used to help make you a better penetration tester. It pays to know where and how your actions will end up in logs. In addition to keeping the SIEM VM, consider taking a closer look at some of the incident response and security monitoring tools mentioned in section 18.2.3

Do you want to set up Windows Server VMs for your lab environment? Microsoft offers evaluation copies of Windows 10, and Windows Server releases ranging from 2012, through 2019. Take a look here: <https://www.microsoft.com/en-us/evalcenter/>. Unfortunately finding legal, safe evaluation downloads for older versions of Windows (e.g., Windows XP, Windows 7, Server 2008, etc.) is a much more difficult task that I cannot assist you with.

There are multitudes of other projects and "capture the flag" contests that students can try out. Some of the more well-known ones include:

OverTheWire: <https://overthewire.org/wargames/>

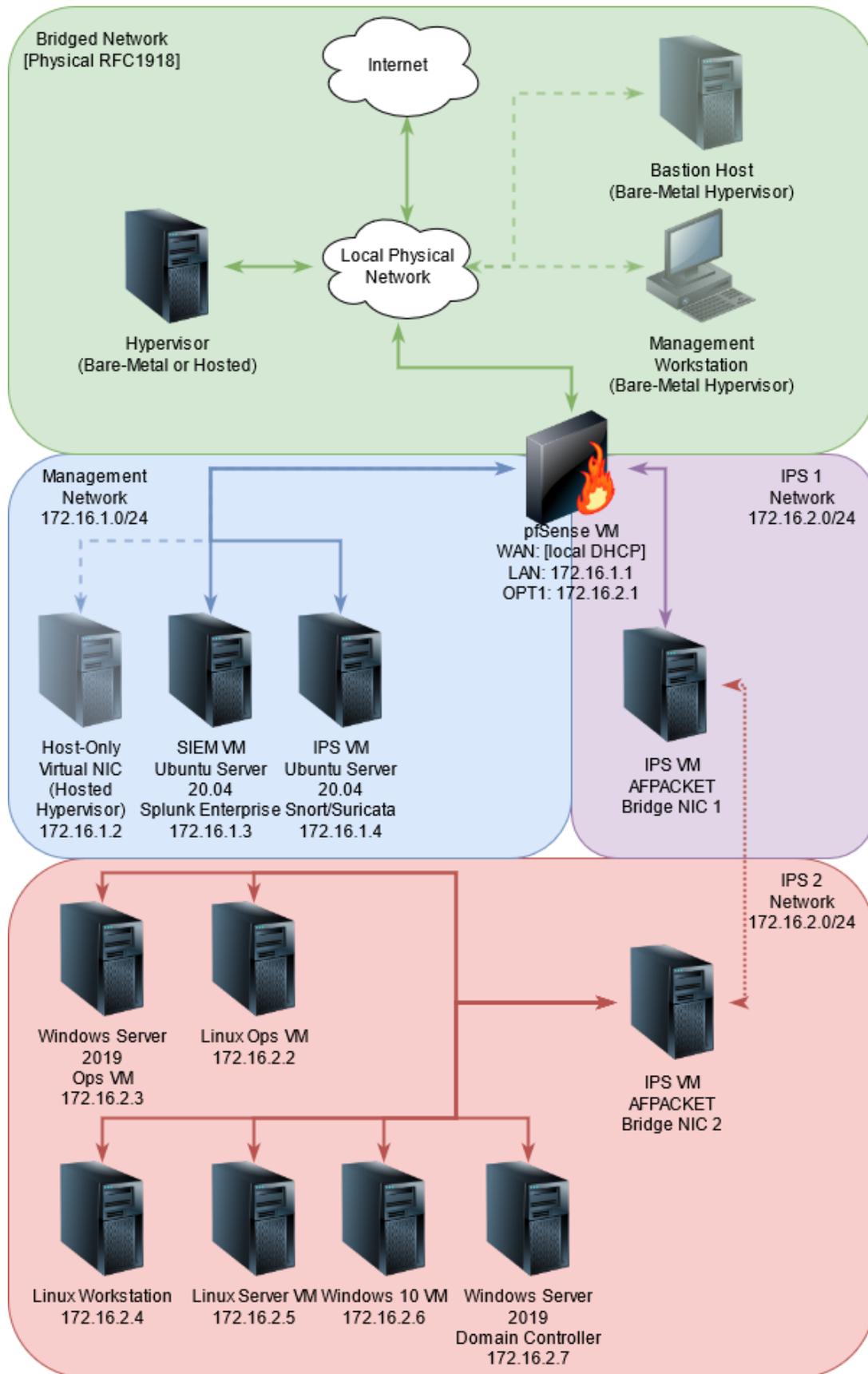
TryHackMe: <https://tryhackme.com/>

HackTheBox: <https://www.hackthebox.eu/login> (note: can be a little tricky)

SANS Holiday Hack: <https://holidayhackchallenge.com/past-challenges/>

As always, these resources are just the tip of the iceberg. Good luck, and have fun.

19.2.5 Outfitting an Ops-Centric lab



Notice in the network diagram that the Kali and Metasploitable 2 VMs are gone, because students don't need them. Other than that, it's no coincidence that the ops lab mockup is similar to the penetration testing lab mockup in the previous section. Pentesting labs require a variety of virtual machines in order to learn how to enumerate, exploit, and pivot across a variety of systems, running various applications and services. In comparison, aspiring system administrators should have access to a variety of systems in order to learn how to manage them all more effectively. These days, systems administration encompasses a wide variety of tools and software for monitoring (e.g. Nagios, Icinga, OSQuery, Spiceworks, SolarWinds, Splunk, etc.) in addition to pushing software packages (e.g. Linux package repository mirrors, Linux Kickstart, Microsoft System Center, Windows Software Update Deployment Services, etc.), DevOps tools to automate tasks (Terraform, Ansible, Chef, Puppet, Salt, Vagrant, etc.), and container technologies (Docker, LXC, Kubernetes, Packer, etc.), to say nothing of the various services and applications that run on different operating systems and cloud providers.

The goal of this lab environment is to provide a solid mix of both Linux and Windows virtual machines to learn how to use various cross-platform, (as well as platform-specific) administration tools and software more effectively. In terms of resource allocations (again, assuming 1GB of RAM per Linux VM in the IPS2 network, and 4GB of RAM for each Windows VM, and 40GB of disk space per virtual machine), total resource allocations (including the SIEM, IPS, and pfSense VMs) come to 23.5 GB of RAM, and at least 405GB of disk space. As usual, these figures are estimates that do not include VM snapshots, and/or operating system overhead. I would recommend a system with at least 32GB of RAM and 800GB of free disk space.

As always, if the resource requirements are too high to run the full lab environment, students should consider focusing on studying just the subjects they are immediately interested in, allocating only the resources they need. Interested in learning Linux administration? Drop the Windows boxes for now. Need to learn how to set up a Microsoft network? Consider dropping the Linux systems for the time being. Interested in learning BSD instead? Scrap all the VMs and install just the BSD virtual machines you require. Tune the environment to fit your needs.

Kickstarting Your Ops Lab

In my opinion, systems administration (and by extension, yes, that includes devops) has a much broader scope and depth to cover than either malware analysis or penetration testing. As such, it's a little bit more difficult for me to recommend solid resources, since goals are going to differ in terms of what systems administration functions students want to learn, as well as the platform they want to learn it on

Many of the same resources I recommended in chapter 2 are a tremendous boon towards learning how to navigate the Unix/Linux command-line, cybersecurity, and computer networking in general. However, when it comes to specific technologies, I don't have very many hard, fast recommendations, check out the sidebar conversation below, *Obtaining the Guidance You Seek*, for some recommendations on finding resources to best suit your needs.

Obtaining the Guidance You Seek

At this point, I've provided you with a couple of network diagrams that demonstrate how the baseline lab environment can be retrofitted to better serve your needs – be they blue team endeavors, red team endeavors, pursuits into devops/administration, etc. Along with each of these network diagrams, I provided a couple of recommendations for specific software, tools, and/or training materials that I thought might help students toward their goals. However, Systems Administration, and Information Security are pretty broad topics, and there may be a chance that the resources I pointed you towards, or the tools I recommended may not cover a particular subject, technique, or discipline you're interested in. Here are a couple of recommendations that may help you find the resources and guidance you seek:

See if the skill, software, or operating system you want to learn more about has some sort of a certification track. Now, I'm not going to advocate for or against acquiring certifications, because that's a can of worms in and of itself, but I will say that many of the textbooks that are geared towards helping to obtain specific certifications are going to be your best bet towards learning the core concepts. Will they teach you every single facet? Probably not. Will you finish a book knowing more than what you started with? Almost certainly. The only downside to using certification textbooks is that sometimes the material can be a little cut and dry, especially if there are no lab exercises interspersed in with the material. But the benefit of having a VM lab means you can get hands-on with the material while you learn.

See if the subject you want to learn more about has an online community of some sort. If you join an online community dedicated towards the subject you're interested in, the members may have compiled a "getting started" list of resources that they point new members to, or they might have recommendations to offer if you post in their communities asking for help. Before doing so however, let me offer another significant piece of advice:

If you elect to join these online communities, be patient when asking for resources. Netiquette describes acceptable ways of communicating on the internet, and is very important towards acquiring the information and resources you want. For the most part netiquette boils down to: clear communication, treating others how you want to be treated, and being patient – especially if you do not get a response to your requests for help as fast as you would like. Remember that those strangers out there are volunteering their free time to whatever online community you joined or decided to frequent. Put yourself in their shoes, and wait it out for a little bit.

Another recommendation is to read the work, "How to Ask Questions the Smart Way":

<http://catb.org/~esr/faqs/smart-questions.html>

To summarize, it's more or less an expansion upon the netiquette, including recommendations to try searching for a resolution to your problem on your own, before asking others for help.

At the same time however, **do not tolerate abuse from strangers.** In the olden days of the internet, a popular online web comic called Penny Arcade, featured what has become one of the most well-known theories of internet communication: *The Internet _____wad Theory*. To summarize, relative anonymity plus an audience creates an environment where abusers thrive. Never settle for abusive behavior. Mute, block, or otherwise ignore abusers, especially if they continue to mock you, if you've done nothing to deserve it. Don't let abusers color your impression of that community, unless the problem is systemic, and the moderators/staff of that community are unwilling to do anything. If the problem *is* systemic, consider looking at other online communities, or communities for competing software platforms instead.

If free resources and/or online communities are a bust, considering searching for classes or other formal training on the subject. I know that this may not be the most affordable recommendation, but if you choose to pursue it, you'll have paid for a professional service, and will have the right to learn in a structured environment free from abuse. If the problems continue after that, you'll also have the right to submit a review, or escalate the issue to authority figures in the organization who offer the training.

If all else fails, grab a virtual machine, grab a software trial if it's available, and take the hands-on approach. Install it on your own, then document your successes and/or failures. Make use of snapshots if configuration changes result in stability problems. This is *your* virtual machine lab, and is designed for *you* to experiment and/or make mistakes in the pursuit of learning.

Special Mention: DetectionLab

I wanted to take a moment to mention a special project called DetectionLab, created by Chris Long. To put it bluntly, DetectionLab is a virtual machine environment all its own, and I think it's even better than the baseline environment I made you put together manually. The project's description states that it's "built with defenders in mind" (e.g., security analysts, incident responders, threat hunters, malware analysts, etc.) but in my opinion, it has something for everyone.

Defenders will enjoy that it has so much logging and so many security monitoring tools built into it. Offensive security professionals will likely enjoy it as a test bed to determine where their actions are logged, or maybe even as an environment for testing new tools, techniques, and tradecraft. Finally, DevOps/Systems Administrators may find the build automation responsible for putting the entire lab together intriguing, and may wish to learn more about the tools used to handle build automation (e.g., Hashicorp tools – Packer, Vagrant and Terraform, as well as Ansible).

If this sounds interesting to you (and it definitely should), the project is available on github at:

<https://github.com/clong/DetectionLab>

If DetectionLab is so great, why have you waited until now, the last pages of the last chapter of the book, to bother saying anything about it?

Because knowing how to do it all on your own manually is still extremely valuable. The journey is at least as valuable as the destination. Sure, I could've pointed you to DetectionLab in the first chapter. It would've made this book much shorter, and I probably would've freed up an entire year of effort, but I made you do it all *manually*. Now, not only will you better appreciate the automation that went into such a great project, you'll be in a better position to support your lab environment, troubleshoot it yourself if something goes wrong once the lab is built, or if something goes wrong during the build process. Not to mention, if you find DetectionLab to be lacking in some feature or another, you'll be capable of expanding it to suit your needs, thanks to the work you've already put in.

19.3 Final Words

As of mine writing this, this re-write has taken almost an entire year to complete. Time well-spent I think, even if I do have a few more grey hairs, and I've added somewhere around 400 pages to the already gargantuan 600 pages of the first edition. What I thought would be a simple update along with maybe a couple of software version and/or illustration changes here and there ended up being a complete re-write of the material. Why? Because I began reviewing the old material, and almost immediately, I knew I could do better. Not only that, those who have been promised that this work will help them get established in information technology deserve a work that more thoroughly assists them. Those who supported me the first time around need to know that this isn't just a cheap money grab, but truly the efforts of someone who wants to make a work they believe in that much better.

On that note, I want to take the time to once again offer my thanks for your patronage. I've had so many people talk to me and tell me about how they shared the first edition with their friends and colleagues. How it helped them get their foot in the door for their first job. I've seen my peers mention to me how their professors added it to their course syllabus. Never in my life would I have *ever* imagined that any work I committed to paper would ever make it on to a college campus!

In closing, I know there are no shortage of books and blog posts on cybersecurity and establishing your own home lab. I hope that you found this work worthy of your time and that it helped you along in your career. I hope that you learned something new along the way, and that this book becomes a dusty reference tome on your bookshelf you pull out once in a while to help remember the finer details of some subject or another. As I said before in the first edition, welcome to Information Technology and/or Information Security. Have fun, but don't burn yourself out. The fight and the struggle will still be there tomorrow. Take the knowledge you gained here, apply it, better yourself, and those around you by sharing it openly. We either all improve together, or we don't improve at all.

Tony "DA_667" Robinson

Chapter 20 Patch Notes

- Welcome to the extra content chapter.
- I decided to move some of the hypervisor hardening hints that I sprinkled in the middle of the first edition into a single chapter.
- The first edition had content for hardening Windows hosted hypervisors only, but this chapter has guidance for hardening hosted hypervisors and/or manager workstations running Windows, Linux and MacOS
- Covered guidance on how to configure a host-based firewall from MacOS (Murus) Linux (gufw) and Windows (wf.msc)
- Also provided guidance on the updater script – a small shell script for patching the Ubuntu/Kali virtual machines and rebooting them automatically
- Guided students on how to place the script into `/etc/cron.(daily|weekly|monthly)` to be processed by run-parts, or make their own cron job in `/etc/crontab` to have more granular control as to when the script runs and reboots the virtual machine.
- Briefly mentioned systemd-timers, yet another thing systemd has seen fit to re-invent. With thanks to twitter user, @MyArashiyama
- Moved guidance on how to install ntpd on the SIEM, IPS and kali VMs, and configure it to use the NTP service on the pfSense VM.

Chapter 20: Extra Credit

You're still here?

It's over.

Go home.

Oh. You're expecting the extra content I said I'd deliver. Be aware that this chapter is going to be very informal. Let's get started.

20.1 Hardening Hypervisor Security

Regardless of whether or not students are using a hosted or bare-metal hypervisor, segmentation and security are extremely important, not only from the perspective of the virtual lab itself, but for the hypervisor as well. Here are some recommendations to follow in order to help keep the hypervisor (bare-metal or hosted) secure:

Keep the hypervisor and/or hypervisor host patched. I know that this is one of those "Well no duh" recommendations, but I'm gonna say it anyway: If running a hypervisor of any sort, hosted or bare-metal, keep up with the latest releases and install them as soon as possible. As for hosted hypervisors, keeping the host operating system and its applications updated helps to reduce your overall attack surface as well. This guidance also applies to bastion hosts and management workstations for bare-metal hypervisors – yeah, I'm watchin' you. Nobody wants to reboot for Windows Update, or spend an hour waiting for Apple to let you have your computer back while installing the latest macOS updates, but here we are.

Install and configure a host firewall. Again, most student will look at this and say duh, but I have some more specific recommendations to offer:

MacOS Hosts: Technically, buried somewhere deep within MacOS, the pf firewall is buried and lies dormant. And based on others experiences, enabling it and getting any custom firewall rules to stick with it is an exercise in pain, at least not without some help.

Murus is a graphical front-end to the pf firewall that'll make things a little bit easier. Students can download the lite edition for free, while the Basic edition is 10.00 USD, and the Pro edition is 35.00 USD. I recommend picking up the Basic or Pro editions, because unfortunately, the ability to create custom firewall rules is gated behind the paid editions, and in order to get pf to play nice with any sort of consistency, Murus is invaluable.

Either enable pf manually (if you have the insanity or experience to do so) or install Murus, and create the following rules:

Allow outbound connections from 172.16.1.2 to 172.16.1.0/24. Be sure to set the "Keep State" option.

Allow outbound connections from 172.16.1.2 to 172.16.2.0/24. Be sure to set the "Keep State" option.

Deny inbound connections from 172.16.1.0/24

Deny inbound connections from 172.16.2.0/24

If students are using an alternate network configuration for their lab environment, be sure to substitute 172.16.1.2 with the IP address assigned to either vboxnet0 or the vmnet2 host-only network adapter. Additionally, substitute the subnets 172.16.1.0/24 and 172.16.2.0/24 with the subnets for the Management and IPS networks, respectively.

Alternatively, if students are using MacOS as their management workstation to connect to a bastion host in order to access their lab environment on a bare-metal hypervisor, create the following rules instead:

Allow outbound connections from your management workstation to the [IP address of the bastion host]. Be sure to set the "Keep State" option.

Deny inbound connections from [IP address of bastion host]

The easiest way to implement the firewall rules above is via Murus, and its *Custom Rules* feature. Unfortunately, if students want to use Murus to set up their firewall, this feature requires a Basic license or better. Select the *Custom Rules* option under *Configuration* on the left most pane, then in the center pane, Click the + button under *Custom Rules*, and select the option *Add Custom Filtering Rules Container*. This will create an entry in the center pane labeled *Custom Filtering Rules*. Click on it, and in the pane on the right click the + button that appears. A new window labeled *New Custom Filtering Rules custom rule* will appear.

I'll walk students through creating their first block, and first pass rule, then leave it as an exercise to configure the second block and second pass rule on their own. Begin by locating the option labeled *Action*, click *Pass*, and in the drop-down menu that appears, select *Block*. Next, locate the *Direction* option, and in its drop-down menu, select *Inbound*. Afterwards, locate the *Source* option, and in the drop-down menu, select *Address*. An input box appears to the right of the *Source* option. Input 172.16.1.0/24 into the input box, and hit enter. As always, if students are using a different network range for the Management/LAN network for their virtual lab, substitute as necessary. If students did everything correctly, the *Rule preview* section should read:

block in from 172.16.1.0/24 to any

Click the *Add New Rule button* in the lower right corner to finish. Students will be taken back to the *Custom Rules* option, and the *Custom Filtering Rules* container in the middle pane should be selected. On the right pane, a new firewall rule should be visible, labeled: *block in from 172.16.1.0/24 to any*. Repeat the process again, adding a second block rule, making the Rule preview section read:

block in from 172.16.2.0/24 to any

Click the *Add New Rule* button once more, and the third pane next to the Custom Filter Rules container should read:

block in from 172.16.1.0/24 to any

block in from 172.16.2.0/24 to any

That should do it for block rules. The next thing students need to do is add stateful allow rules that allow the host-only network adapter's IP address to access 172.16.1.0/24, and 172.16.2.0/24. Again, if students are using a customer network configuration for their lab environment, substitute as necessary.

In order to create custom pass rules using Murus, Navigate back to the *Custom Rules* option, under *Configuration* on the left pane. In the center pane, select the *Custom Filtering Rules* container, and click the + button on the right pane once more. Locate the *Action* option and confirm its set to *Pass*. Next, locate the *States* option, then in the drop-down menu, select *keep state*. Next, locate *Source*, and its drop-down menu select *Address*. Enter 172.16.1.2 in the input box that appears to the right of the *Source* option, and press the enter key. Repeat this process for the *Destination* option, and input 172.16.1.0/24 into the input box that appears to the right of the *Destination* option. If done correctly, the rule preview option will read:

pass from 172.16.1.2 to 172.16.1.0/24 keep state

Click the *Add New Rule* button, then repeat the process again. Create a pass rule, ensure *keep state* is selected, set the source IP address to 172.16.1.2, and the destination to 172.16.2.0/24, and add that rule to the *Custom Filtering Rules* container. When students are finished adding all of their block and pass rules, the third pane should look like this:

block in from 172.16.1.0/24 to any
block in from 172.16.2.0/24 to any
pass from 172.16.1.2 to 172.16.1.0/24 keep state
pass from 172.16.1.2 to 172.16.2.0/24 keep state

When finished, students will need to click the play button located at the top of the PF Firewall Configuration window to reload the pf firewall with these new rules.

What about students who are accessing a bare-metal lab through a bastion host? Let's assume the bastion host for the bare-metal lab environment is assigned to the IP address 10.0.0.7. Create a custom block rule that reads:

block in from 10.0.0.7 to any

followed by a pass rule that reads

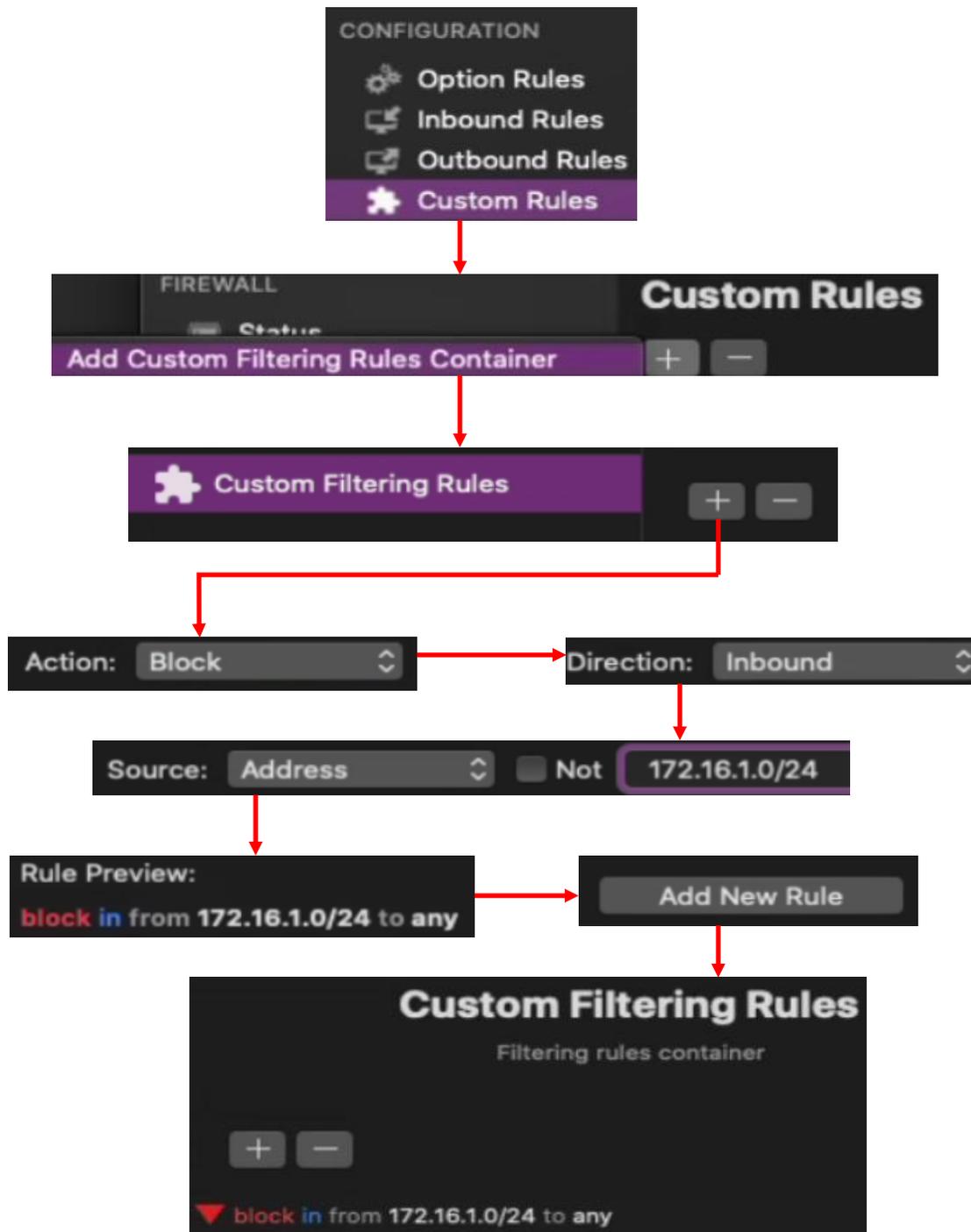
pass from any to 10.0.0.7 keep state

That third pane to the right of the *Custom Filter Rules* container will read:

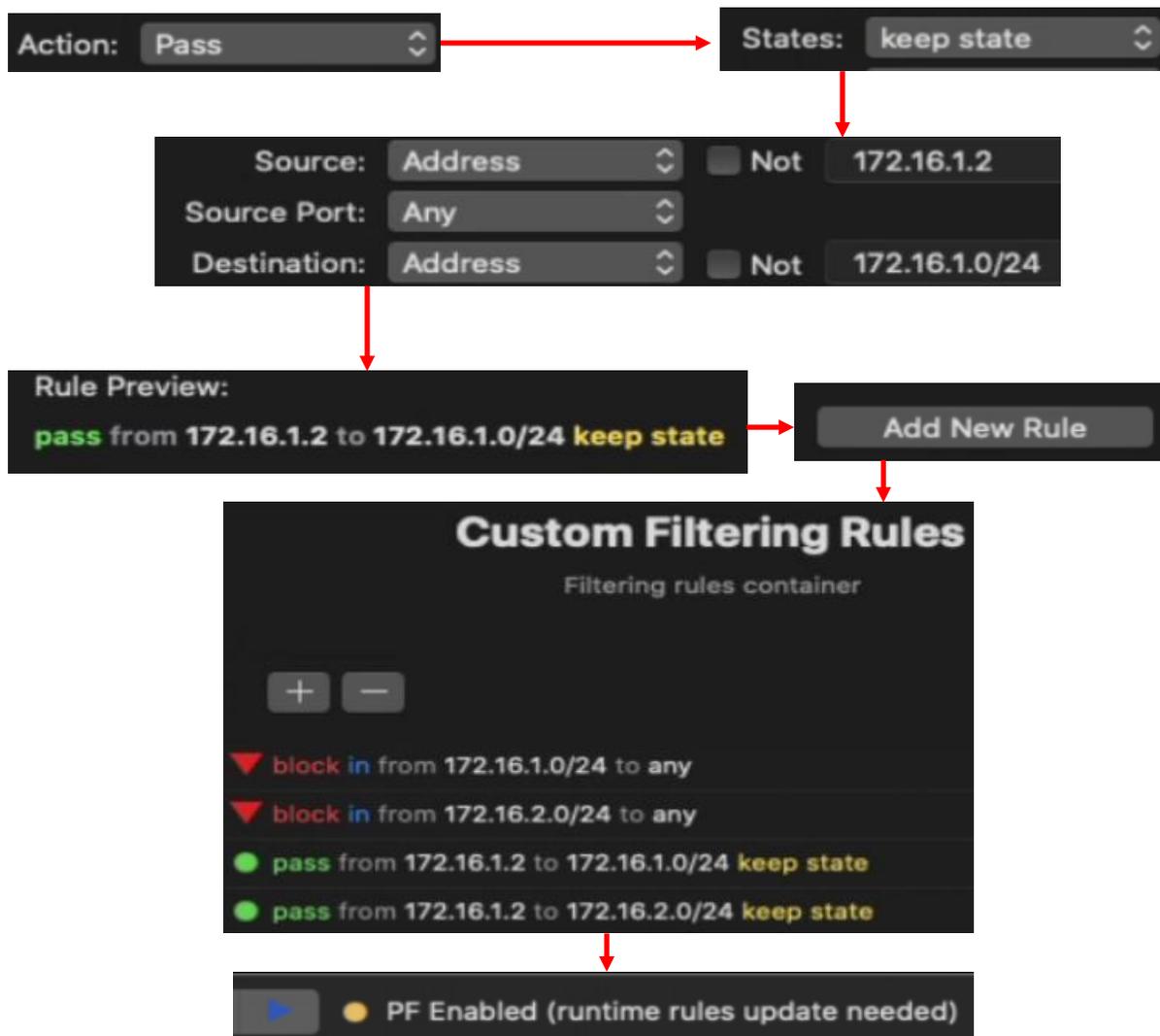
block in from 10.0.0.7 to any
pass from any to 10.0.0.7 keep state

Click the play button to restart the firewall, and that should be all there is to it.

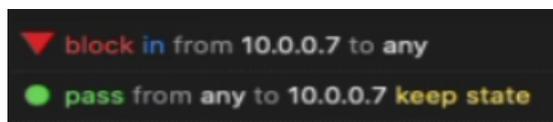
Note: Murus firewall enables a default Inbound Deny firewall rule policy. However, Murus is also able to identify what network services, if any, are running on your mac. From there, you can configure the inbound firewall policy to allow access to specific services.



20-1: Murus can be very complex, but I promise you, it's much better than trying to activate and configure pf on MacOS without it. The easiest way to add our firewall rules is via the Custom Rules option available in Murus Basic or Premium editions. In the *PF Firewall Configuration* menu, click on the *Custom Rules* menu option under *Configuration* on the left pane. Then in the middle pane, click the + button, and select *Add Custom Filter Rules container*. Select the option labeled *Custom Filtering Rules* in the center pane, then on the right-most pane, click the + button to open a new window titled *New Custom Filtering Rules custom rule*. Ensure that *Action* is set to *Block*, *Direction* is set to *Inbound*, and that the *Source* is set to *Address*. In the input box that appears to the right of the *Source* field, input 172.16.1.0/24, and hit enter. Finally, click the *Add New Rule* button. Repeat this process once more, adding another inbound block rule, where the *Source* is set to 172.16.2.0/24.



20-2: Students will need to add two additional rules to allow connections from the host-only network adapter's IP address (e.g., 172.16.1.2) to the Management and IPS1/2 network ranges (e.g., 172.16.1.0/24, 172.16.2.0/24). The process is practically identical to the one used to create the block rules. For the first pass rule, set the *Action* to *Pass*, the *States* option to *keep state*, the *Source* option to *Address*, entering 172.16.1.2 into the input box, and Finally set the *Destination* to *Address*, and enter 172.16.1.0/24. When finished, Click the *Add New Rule* button. Repeat this process once more, creating a new pass rule with keep state set, from source 172.16.1.2 and destination 172.16.2.0/24. When students are finished, the third pane, labeled *Custom Filtering Rules*, should have four rules in total. The rules should be in the exact order with the exact configuration depicted in the illustration above. If students are using an alternate IP address for the hypervisor host's host-only network adapter, and/or for the management and IPS1/2 network ranges, please substitute as necessary. When finished, click the play button at the top of the window.



20-3: On the other hand, if students are using MacOS as a management workstation and connecting to a bare-metal lab through a bastion host, that only requires two firewall rules: one blocking inbound access from the bastion host's IP address and a secondary rule allow network connectivity to the bastion host's IP address, with the *keep state* flag set. In the illustration above, I created a separate set of rules, with 10.0.0.7 serving as the IP address of the bastion host.

Linux Hosts: There are as many Linux distributions as there are stars in the sky. Some are simple clones of their peers with a facelift, and others are alien technology brought to the masses. Every distro is a world unto itself. Not unlike how each distribution is sovereign, the software used to operate the firewall is itself sovereign, and arcane. Here is the state of Linux host-based firewalls, as I understand it:

At the end of the day, most firewall software interacts with `iptables` in some way, shape, or form. So, if you don't like how your distro manages the firewall, figure out how to tear it out, and just use the `iptables` suite of commands. I will not be covering how to do that in detail, because I'm extremely lazy, but the option is available.

UFW, shorthand for uncomplicated firewall is the default firewall manager for most Debian-based distros, should you choose to operate a firewall. It's supposed to be a "human-friendly" front-end to `iptables` to make it easier to work with in general. UFW has a graphical front-end tool available for configuring the firewall, called `gufw`.

Firewalld on the other hand, is considered to be the rough equivalent to UFW, except for Redhat and its derivatives – a front-end for `iptables` meant to make things easier. *Firewalld*, just like UFW has a graphical front-end tool available to ease configuration called `firewall-config`.

If you're reading this, and you're already proficient enough with `iptables` that you've ripped out *UFW*/*Firewalld* in favor of doing things yourself, you'll want the following rules:

On the INPUT chain:

allow RELATED and ESTABLISHED connections

Drop connections from 172.16.1.0/24

Drop connections from 172.16.2.0/24

On the OUTPUT chain:

Allow connections from 172.16.1.2 to 172.16.1.0/24

Allow connections from 172.16.1.2 to 172.16.2.0/24

As always, the IP address 172.16.1.2 is the default IP address for the Linux hypervisor host's host-only network interface (e.g., `vboxnet0` or `vmnet1`). Additionally, 172.16.1.0/24 is the default network range for the Management network, while 172.16.2.0/24 is the default network range for the IPS1 and IPS 2 network segments. If you're using an alternate network configuration to avoid network conflicts, substitute IP addresses and ranges as necessary.

As for the rest of us mere mortals, I recommend using one of the graphical front-ends instead. I've had a chance to experiment with both `firewall-config`, and `gufw`, and in general, I feel like `gufw` is much easier to work with, so if it's within your power, and you want to follow along with the rest of this section, install `ufw`, and use `gufw` to manage your Linux hypervisor host's firewall.

As most Linux applications that change how the system fundamentally operates, *gufw* requires root or *sudo* permissions to do anything of value.

Begin by starting *gufw*. The default screen most students will see is a window labeled *Firewall*, along with word *Firewall* and a host of menu options – *Profile*, *Status*, *Incoming* and *Outgoing*. Ignore the *Profile* setting for now, and just accept whatever profile *gufw* uses for assigning these firewall rules. In my case, I'm using the *Home* profile. Next up, make sure that the *Status* option is set to on. This can be done by clicking the slider button towards the right. If done correctly, the shield icon will light up and turn green, white and red. Next up are *Incoming* and *Outgoing*. As the names imply, these are the default configuration settings for the firewall for incoming network connections, as well as outgoing network connections. The default configuration of *Incoming* set to *Deny*, and *Outgoing* set to *Allow* is sensible. However, if students are running any special services on their hypervisor host, they will be blocked until either the *Incoming* policy is set to *Allow*, or firewall rules are configured to allow connections to specific ports.

If students aren't running any special network services on their hypervisor host, and don't care to host any network services on it in the near future, then this is quite literally all students need to make the hypervisor host more secure: The host is allowed to connect outbound to the lab environment, while the lab environment is denied access to the hypervisor host over the network. However, if students are running network services on their hypervisor host, then the default configuration of denying network traffic inbound is going to cause problems. There are two possibilities here:

- Keep the default *Inbound Deny* configuration, but create allow rules for each of the services you want others to have access to
- Switch the *Inbound* configuration to *Allow* and create custom firewall rules to block access from the lab network ranges specifically

Regardless of the choice students make here, I recommend creating additional firewall rules to enforce denying traffic inbound from the lab environment, while allowing traffic outbound from the host-only interface to the lab environment. Here's how to do that:

Click on the button near the bottom labeled *Rules*, then in the blank window pane that appears, click the *+* button. This opens a new window labeled *Add a Firewall Rule*. Click the option at the top of the window labeled *Advanced*. There are a lot of options on this screen, but don't worry, we'll get through it. For your first firewall rule, let's create a pass rule from 172.16.1.2, to 172.16.1.0/24. Edit the following fields:

Name	Management Allow
Policy	Allow
Direction	Out
Protocol	Both
From	172.16.1.2
To	172.16.1.0/24

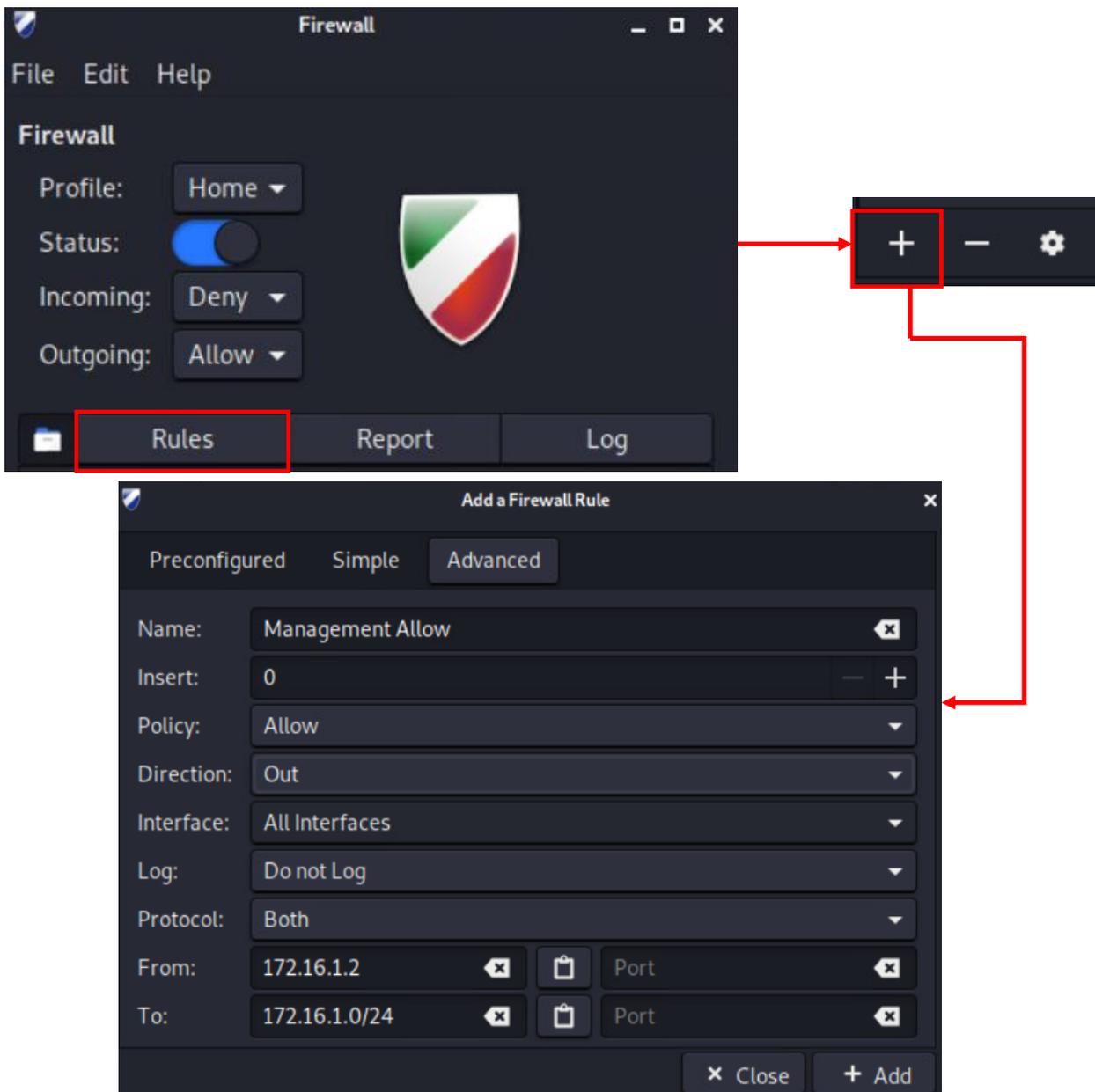
When finished, click the *Add* button in the lower right corner, then close the window. The new firewall rule will appear in the previously blank window under the *Rules* option. Now, students need to add 3 more firewall rules to the firewall by repeating this process three more times. Here are the settings to use:

Name	IPS Allow	Management Deny (In)	IPS Deny (In)
Insert	1	2	3
Policy	Allow	Deny	Deny
Direction	Out	In	In
Protocol	Both	Both	Both
From	172.16.1.2	172.16.1.0/24	172.16.2.0/24
To	172.16.2.0/24	[blank]	[blank]

Some students may note that the firewall rules don't mention keeping state (e.g. there is no `-m state --state RELATED,ESTABLISHED` rule inbound). That's because `ufw` creates that firewall rule *implicitly*. Bottom line is that students don't need to worry about that rule when using `gufw`.

Alternatively, if students are using their Linux system as a management workstation, and connecting to a bare-metal hypervisor lab through a bastion host, only two firewall rules are needed. Remember that the bastion host's IP address will vary depending on the network range students use for the local, physical network. Enter the following firewall rules below, substituting 10.0.0.7 with the IP address of the bastion host system as necessary:

Name	Bastion Host Allow	Management Deny (In)
Insert	1	2
Policy	Allow	Deny
Direction	Out	In
Protocol	Both	Both
From	[blank]	10.0.0.7
To	10.0.0.7	[blank]



20-4: Run `gufw`, and on the first screen that appears, click the *Status* switch to the on position (towards the right). I recommend leaving the remaining settings defaults. Next, click the *Rules* option, then at the bottom of the window, click the `+` button to add a new firewall rule. Set the *Name* to *Management Allow*, the *Policy* to *Allow*, the *Direction* to *Out*, and the *Protocol* to *both*. Finally, in the *From* field set the IP address `172.16.1.2`, and in the *To* field, set the IP address to `172.16.1.0/24`. When finished, click the *Add* button. If students are using a non-default set of IP addresses and network ranges for their lab environment, substitute as necessary.

N°	Rule	Name
1	172.16.2.0/24 ALLOW OUT 172.16.1.2 (out)	IPS Allow
2	Anywhere DENY IN 172.16.1.0/24	Management Deny (In)
3	Anywhere DENY IN 172.16.2.0/24	IPS Deny (In)
4	172.16.1.0/24 ALLOW OUT 172.16.1.2 (out)	Management Allow

20-5: Using the table on p. 1063, create three additional firewall rules – one more allow rule from 172.16.1.2 to 172.16.2.0/24, and two additional block rules from 172.16.1.0/24, and 172.16.2.0/24.

N°	Rule	Name
1	10.0.0.7 ALLOW OUT Anywhere (out)	Bastion Host Allow (out)
2	Anywhere DENY IN 10.0.0.7	Bastion Host Deny (In)

20-6: For students interacting with a bare-metal hypervisor through a bastion host, try out these firewall rules instead. Substitute 10.0.0.7 with the IP address of the bastion host, as necessary.

Windows Hosts: Windows Firewall has a New Rule wizard that is pretty easy to add new rules to. The easiest way to access the *Windows Defender Firewall with Advanced Security* application is to open a Windows run prompt (*Start > search for the Run app*, or alternatively, the windows meta key + R), type in `wf.msc`, then hit enter.

By default, outbound connections that don't have a firewall rule are allowed, while inbound rules that don't have a matching rule are blocked. This is almost identical to how the firewalls operate on both MacOS and Linux. For the sake of being sure however, we'll be creating four firewall rules. Two rules to allow access from 172.16.1.2 to 172.16.1.0/24, and 172.16.2.0/24 on the *Outbound Rules* chain, and two rules that deny inbound connection from 172.16.1.0/24 and 172.16.2.0/24 on the *Inbound Rules* chain. Let's begin with the inbound rules.

In the Windows Defender Firewall with Advanced Security window, click on *Inbound Rules* on the left pane. Then in the right pane, under *Actions*, click *New Rule*. This opens a new window, and starts the *New Inbound Rule Wizard*. The first screen asks what type of rule students want to create. Click the *Custom* radio button and click *Next*. On the next screen, ensure the *All programs* radio button is selected, then click *Next*. The next screen asks which ports and protocols apply to this rule. Use the default *Protocol type* setting of *Any*, and click *Next*. The next screen asks what local and remote IP addresses this rule applies to. For the local IP address setting, ensure that the *Any IP address* radio button is selected. As for the remote IP address setting, click the *These IP addresses* radio button, then click the *Add* button to the right. A window labeled IP Address pops up. Ensure that the *This IP address or subnet* radio button is selected, and enter 172.16.1.0/24, then click the *OK* button. Repeat this process once more – Click the *Add* button, enter 172.16.2.0/24, then click *OK*. When finished, the *These IP addresses* pane should read:

172.16.1.0/24
172.16.2.0/24

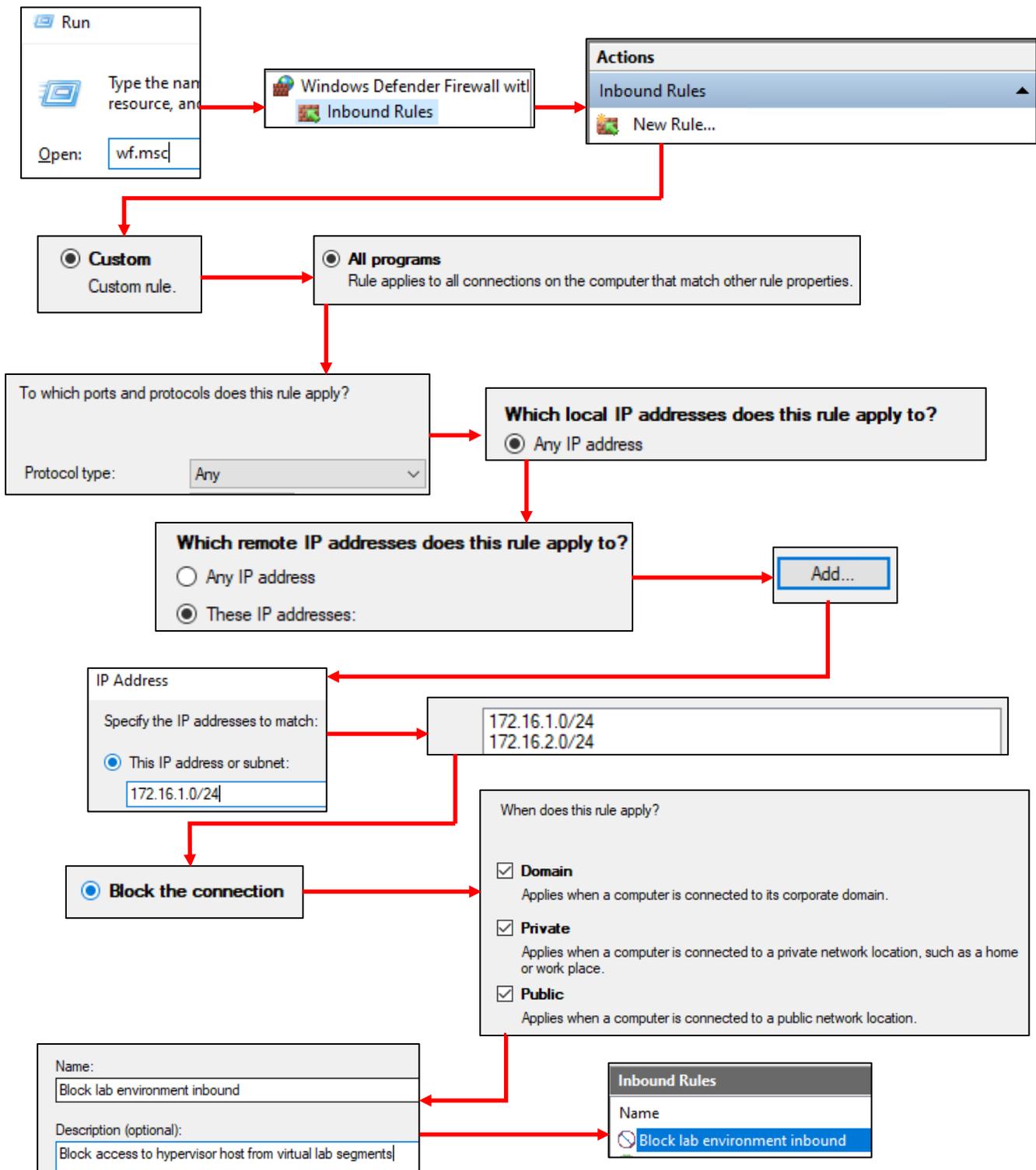
Click *Next* to proceed. The next screen wants to know what to do with network traffic that meets the rule criteria. Click the *Block the connection* radio button, then click *Next*. On the *When does this rule apply* screen, ensure that the *Domain*, *Private* and *Public* checkboxes are all checked – we want this rule to apply at all times, to all profiles. Click *Next*. On the final screen, provide a name and brief description for this firewall rule. For example, I named mine "Block lab environment inbound" and described it as "Block access to hypervisor host from virtual lab segments." Finally, click the *Finish* button to create this rule. The new rule should appear in the middle pane at the top of the *Inbound Rules* chain.

We have one more rule to add, but this time it's to the *Outbound Rules* chain. Click on *Outbound Rules* on the left-most pane, then on the right-most pane, click *New Rule* to start the *New Outbound Rule Wizard*. For the most part, this wizard is identical to the wizard we used to create our inbound rule. Here are the settings to use:

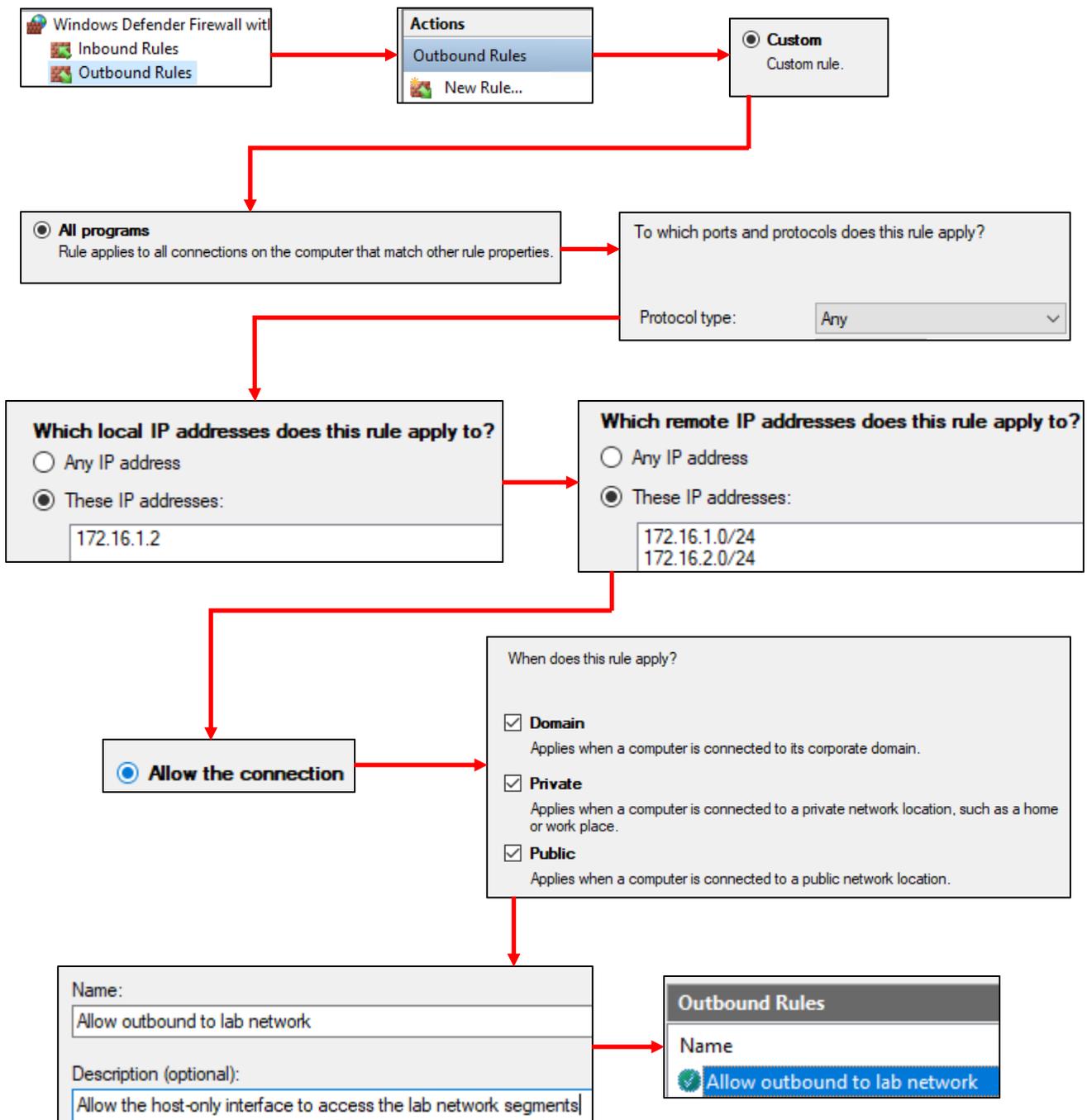
Type of Rule	Custom
Applies to	All programs
Protocol type	Any
Local IP addresses	172.16.1.2
Remote IP addresses	172.16.1.0/24, 172.16.2.0/24
Action	Allow
Profile	All
Name	Allow outbound to lab network
Description	Allow the host-only interface access to the lab network segments

What about students who are using their Windows system as a management workstation, interacting with a bare-metal lab environment through a bastion host? You'll need to create one Inbound and one Outbound firewall rule to allow outbound access to the IP address of your bastion host, and deny inbound access from the bastion host IP address. Check out fig. 20-9 below for more details. Be sure to substitute the IP address 10.0.0.7, with the IP address of the bastion host system as necessary.

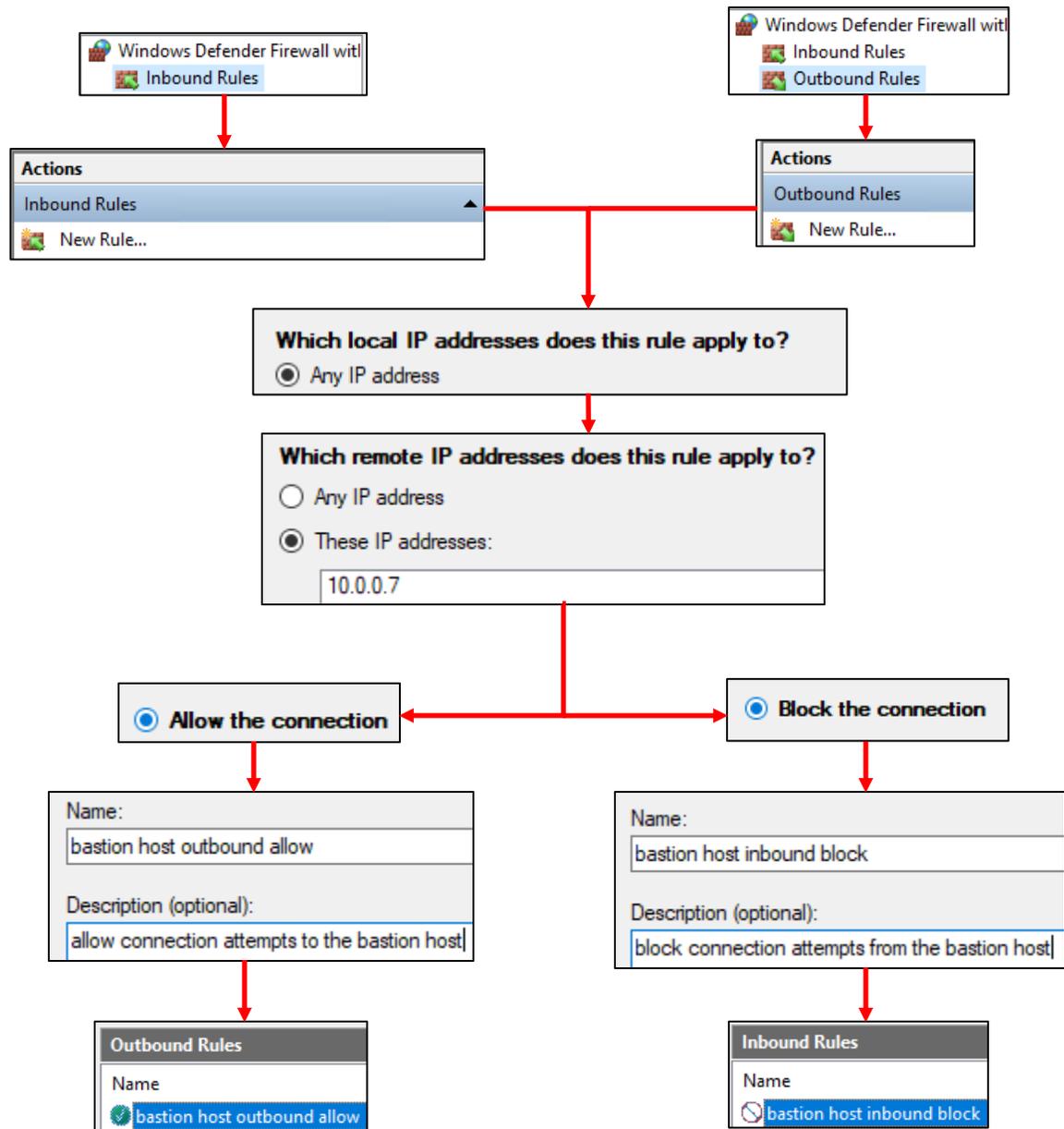
Note: As always, the IP address 172.16.1.2 is the default IP address for the Windows hypervisor host's host-only network interface (e.g., the hyper-v management vswitch, virtualbox host-only network adapter, and/or vmware's vmnet1 network adapter). Additionally, 172.16.1.0/24 is the default network range for the Management network, while 172.16.2.0/24 is the default network range for the IPS1 and IPS 2 network segments. If you're using an alternate network configuration to avoid network conflicts, substitute IP addresses and ranges as necessary.



20-7: Access the Windows firewall, select Inbound Rules from the left pane, then click New Rules on the right pane to start the firewall rule wizard. The illustration above guides students through the wizard, showing the necessary settings to configure a firewall rule that blocks any connections inbound from the 172.16.1.0/24 and 172.16.2.0/24 networks using any network profile.



20-8: The process for creating an outbound allow rule that allows connections from 172.16.1.2 to networks 172.16.1.0/24 and 172.16.2.0/24 is more or less the exact same process we followed for creating the inbound block rule. The only major differences are that students need to select the *These IP addresses* radio button for the local IP addresses configuration option, and specify 172.16.1.2 (via clicking the *Add* button), and ensure that the firewall action is set to *Allow the connection*.



20-9: The process for configuring inbound/outbound firewall rules for controlling access to bastion host(s) is more or less identical to the hosted hypervisor rule process laid out above. I laid out which portions of the *New Firewall Rule Wizard* differ. Assume that all other configuration options should be left at their default settings.

Bonus lesson: Unbinding Windows network protocols from the host-only adapter

Wanna see something fun? Open a command prompt and run the command:

```
netstat -ano | findstr 172.16.1.2
```

If necessary, substitute 172.16.1.2 with the IP address you assigned to the host-only adapter. You'll notice that there are a number of network services listening and that most of them are related to Windows networking nonsense (e.g., 139/TCP, as well as 137/UDP and 138/UDP). I'm going to show you how to disable this nonsense as an added layer of security for Windows workstations.

To begin open the run prompt, type in `ncpa.cpl` and hit enter to bring up the network control panel. Locate the host-only network adapter you are using to access your hosted lab environment. For demonstration purposes, and because I'm extremely lazy, I will be demonstrating with Client Hyper-V and configuring the *vEthernet (Management)* interface. Locate your host-only virtual network adapter, right-click on it, and select *Properties*. In the *This connection uses the following items* pane, disable all of the following items:

- Client for Microsoft Networks
- File and Printer Sharing for Microsoft Networks
- Microsoft LLDP Protocol Driver
- Internet Protocol Version 6 (TCP/IPv6)
- Link-Layer Topology Discovery Responder
- Link-Layer Topology Discovery Mapper I/O Driver

This is only part one of making your Windows host-only interface as silent as a fart on the first date. Next, locate Internet Protocol Version 4 (TCP/IPv4) setting, highlight it and click the *Properties* button. In the *Internet Protocol Version 4 (TCP/IPv4) Properties* window, click the *Advanced* button in the lower right corner. This brings up the *Advanced TCP/IP Settings* window. Click on the tab labeled *WINS*, and locate the section labeled *NetBIOS setting*, and click the *Disable NetBIOS over TCP/IP option*. We're done messing around with the network interface and the network control panel, so click *OK*, then *OK* again, then click *Close* to exit out of the interface properties, then close the network control panel.

Finally, open up the run prompt again, type `services.msc`, then hit enter. Now, the service menu is dangerous territory, and generally speaking, unless you know what you're doing, mucking about with services and their status on Windows is best left to wizards. But we're going to make a single exception today. Locate the *SSDP Discovery Service*, right-click on it, and select *Properties*.

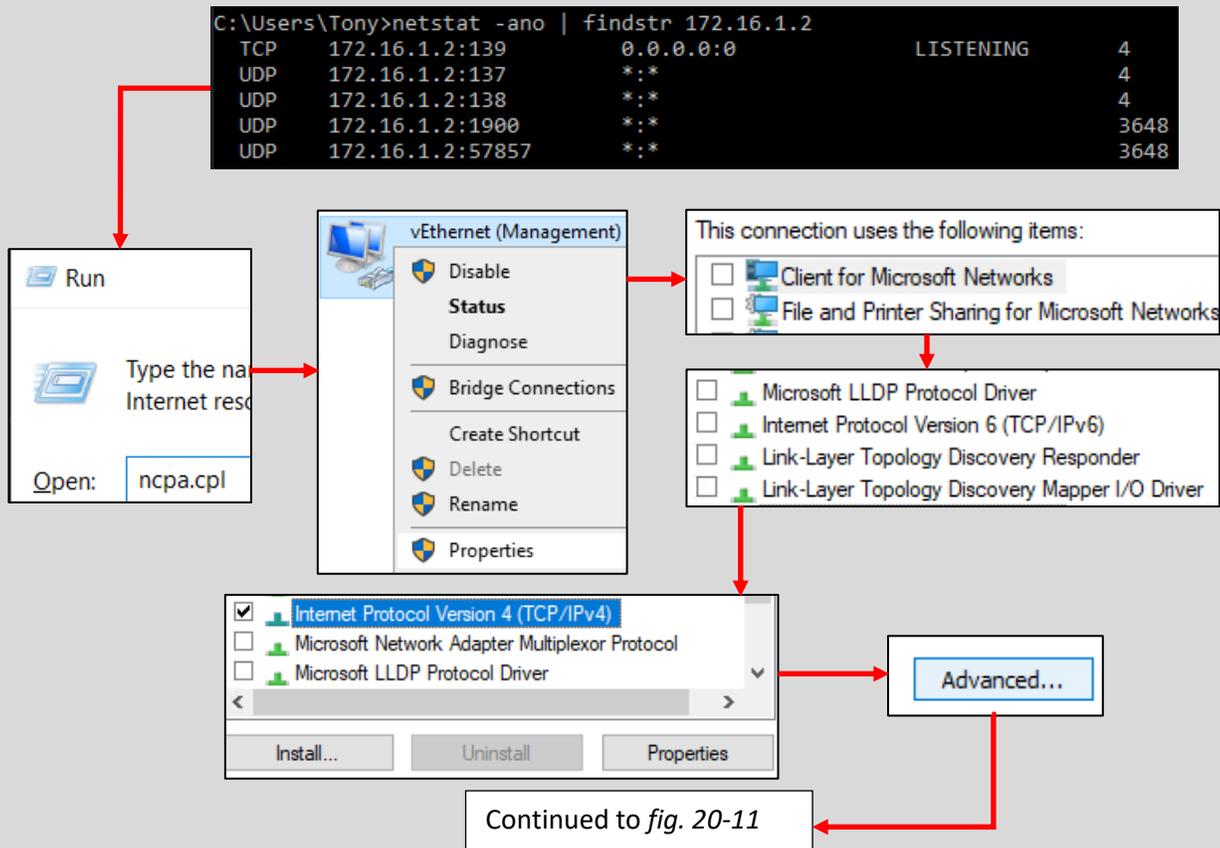
Locate the *Startup type* drop-down setting, and set it to *Disabled*. Then, under the *Service status* section, click the *Stop* button. The *Service status* should then read *Stopped*. Click the *Apply* button, followed by *OK*.

Now, open up the command prompt again, and run the same command we ran earlier:

```
netstat -ano | findstr 172.16.1.2
```

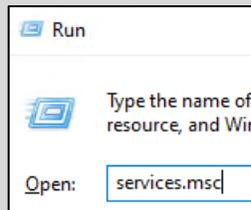
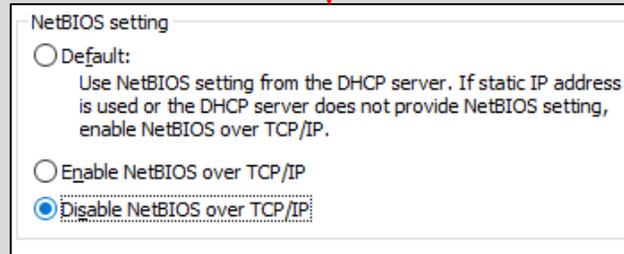
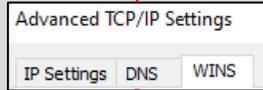
This time around, there are zero network services listening. As far as the virtual lab is concerned, even if the network firewall wasn't blocking connections from the lab segments, there would be no services for any of the virtual machines to interact with on that network interface. Assuming you're not running any other custom network services on your Windows 10 system, or running Remote Desktop Protocol (RDP), in which case, consider adding an inbound block rule to block RDP access from the lab network ranges. Because even if the default firewall configuration is to deny inbound connections, the windows firewall also creates an allow exception when RDP is enabled.

As a final sidenote here, make sure you document changes made to your Windows workstation. Just in case you run into network difficulties later, you'll know to enable the SSDP service again to see if it's the root cause of your configuration problems.



20-10: By default, there are a lot of services listening for connections on all available network interfaces in Windows. The thing is, we don't want any of those services listening on our host-only virtual adapter. To fix this, open up the network control panel (`ncpa.cpl`) and locate the virtual host-only network adapter for your hosted hypervisor, and open its properties. Uncheck the huge laundry list of items above, then open the *Properties* window for *Internet Protocol Version 4 (TCP/IPv4)*. Once in that window, click the *Advanced* button in the lower right corner.

Continued from fig. 20-10



```
C:\Users\Tony>netstat -ano | findstr 172.16.1.2
```

20-11: In the *Advanced TCP/IP Settings* window, click the *WINS* tab, then click the *Disable NetBIOS over TCP/IP* radio button towards the bottom of the window, then click the *OK* button twice, the *Close* button, then exit the network control panel. Next up, open the *Services* manager. The fastest way to do this is to open a run prompt, type in *services.msc*, then hit enter. Locate the *SSDP Discovery* service, then right-click on it, and select *Properties*. Set the *Startup type* to *Disabled*, then click the *Stop* button under *Service status*. When finished, click *Apply*, followed by *OK* and exit the *Services* manager. On the command prompt, students can run `netstat -ano | findstr [host-only interface address]`. If the command returns no output, the configuration changes were successful.

20.2 Update automation with the updater script

The last thing I want to cover is performing update automation for the SIEM, IPS and Kali virtual machines (and/or any Linux VMs students add later, using the `apt` package manager). To do this, we're going to take advantage of a small shell script I've written called `updater`, and the cron job scheduler to do most of the heavy lifting for us. To make a long story short, the script runs `apt-get update`, `apt-get -y dist-upgrade`, then reboots the system. For reasons of liability, this script is MIT licensed, and I would not recommend its use in a mission-critical production environment. The script is available at:

<https://gist.github.com/da667/20f1c67c264f7823c7139f5c835a7026>

to download the script to the SIEM, IPS and Kali VMs, run:

```
export http_proxy=http://172.16.1.1:3128
export https_proxy=
wget
https://gist.githubusercontent.com/da667/20f1c67c264f7823c7139f5c835a7026/raw/42fe847
bbfdd06d24d178d951afff9d8f5cfe4ab/updater
chmod 700 updater
```

The `export` commands are there for the `wget` command to know that there is a proxy that needs to be used for outbound HTTP access. For the Kali VM, be sure to change the `export http_proxy` line to `export http_proxy=http://172.16.2.1:3128`. As usual, if students are using an alternate set of network ranges for the LAN and OPT1 networks (e.g., the Management and IPS networks) substitute as necessary.

After our `export` commands, we use `wget` to download the `updater` script from `gist.github.com`. Finally, the `chmod` command is used to set execute permissions for the `updater` script's owner, and read permissions for everyone else.

For those who are without internet access, below are the contents of the script so that you can transcribe them manually as necessary. When finished, be sure to save the file as `updater` with no file extension. Additionally, be sure to run the command `chmod 700 updater` as well:

Updater

```
#!/bin/bash
#updater.sh - Weekly update script
#checks for updates, downloads them, then reboots the system.
#place this script in /etc/cron.weekly, ensure it is owned by root (chown root:root
/etc/cron.weekly/updater)
#ensure the script execute permissions (chmod 700 /etc/cron.weekly/updater)
#if you want updates to run once daily or monthly, you could also place this script
into cron.daily, or cron.weekly.
#alternatively, edit /etc/crontab to create a crontab entry.
export DEBIAN_FRONTEND=noninteractive
apt-get -q update
apt-get -y -q dist-upgrade
logger updater cron job ran successfully. rebooting system
init 6
exit 0
```

Now that students have the updater script on the SIEM, IPS and/or Kali VMs, perform the following commands:

```
sudo su -
chown root:root /home/[username]/updater
cp /home/[username]/updater /etc/cron.weekly/
```

Now let's run through these commands. As usual, we start with `sudo su -` to become the root user, and also as usual, if students were already the root user when they downloaded/copied the updater script, this command isn't necessary. The next command changes the ownership of the updater file to the root user and group. Finally, we copy the updater script from the home directory of the user who downloaded it to the directory `/etc/cron.weekly`. For these two commands, if students downloaded or copied the script as the root user, substitute `~/updater` for the root user's home directory as opposed to `/home/[username]/updater`.

Placing the updater script into the `/etc/cron.weekly` directory ensures that the script gets ran once per week. However, there are also the `/etc/cron.hourly`, `/etc/cron.daily`, and `/etc/cron.monthly` directories that run scripts placed in them once an hour, once per day, and once per month, respectively. I wouldn't recommend having this script run once per hour, but if students want it to be ran more frequently, copy the script to `cron.daily` instead. Less frequently? Try `cron.monthly` instead.

Curious to see if the updater script ran? Run the following command:

```
grep updater /var/log/syslog
```

If the script was run recently, students should see results similar to this:

May 3 17:24:42 Kali root: updater cron job ran successfully. rebooting system

```
ayy@Kali:~$ export http_proxy=http://172.16.2.1:3128
ayy@Kali:~$ export https_proxy=
ayy@Kali:~$ wget https://gist.githubusercontent.com/da667/20f
--2021-05-03 19:51:11-- https://gist.githubusercontent.com/d
Resolving gist.githubusercontent.com (gist.githubusercontent.
Connecting to gist.githubusercontent.com (gist.githubusercontent.
HTTP request sent, awaiting response... 200 OK
Length: 745 [text/plain]
Saving to: 'updater'

updater                                     100% [=====
2021-05-03 19:51:11 (39.7 MB/s) - 'updater' saved [745/745]

ayy@Kali:~$ chmod 700 updater
ayy@Kali:~$ sudo su -
[sudo] password for ayy:
root@Kali:~# chown root:root /home/ayy/updater
root@Kali:~# cp /home/ayy/updater /etc/cron.weekly/
root@Kali:~# ls -al /etc/cron.weekly/
total 28
drwxr-xr-x  2 root root  4096 May  3 19:52 .
drwxr-xr-x 163 root root 12288 May  3 19:48 ..
-rwxr-xr-x  1 root root   813 Jul  5 2020 man-db
-rw-r--r--  1 root root   102 Feb 10 2020 .placeholder
-rwx-----  1 root root   745 May  3 19:52 updater
```

```
root@Kali:~# grep updater /var/log/syslog
May  3 17:24:42 Kali root: updater cron job ran successfully. rebooting system
```

20-12: This string of commands downloads my updater script from gist.github.com and changes the permissions to allow it to be executed. Then students become the root user, change the ownership of the updater script to the root user and group and copy it to /etc/cron.weekly, where it will be ran automatically, once per week. Students may also place the updater script in /etc/cron.daily or /etc/cron.monthly to have the update script be ran more frequently, or less frequently. The grep command below depicts what a successful execution of the script looks like in /var/log/syslog.

/etc/crontab: A more direct approach

Way back when I was writing the first edition of this book, I experienced a problem where run-parts and /etc/crontab wasn't running the updater script on Kali Linux: For some reason, the script would not run in /etc/cron.(hourly|daily|weekly|monthly). However, I discovered that bypassing run-parts and just telling cron to run my script directly worked fine. So in the interest of providing you an alternate method of doing things in case the lazy way (my preferred way) fails, Here's how to interpret and modify the file /etc/crontab to run the updater script as necessary.

In a nutshell, cron is the name of the service on Linux/Unix systems that is responsible for running tasks that users want or need to run at a particular time, on a specific schedule. A file that describes when and at what frequency to run tasks is called a crontab. The crontab file for the root user is /etc/crontab. Here is a copy of what the crontab looks on my Kali Linux box:

A sample /etc/crontab

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
#
```

So, let's talk about this mess of symbols and lines of content. First and foremost, feel free to ignore just about anything that has a hashmark/octothorpe (#) in front of it. That is to say, its useful information, but they're just comment lines meant to tell you more about this file. Let's start with the lines:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

These are environment variables that tell the `cron` binary what shell to run its jobs with, and what file system paths to use for locating commands in this file.

Next, below the "Example job definition" comment, are the actual jobs that are scheduled to be run by `cron`:

```
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6 * * 7 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 6 1 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
```

Now, assuming that each of these lines didn't blatantly give away that they are the `run-parts` command, running stuff out of `/etc/cron.(hourly|daily|weekly|monthly)`, you could try to make sense of the comment lines above to understand when and how often these commands get ran, or you can do what I did and use `crontab.guru` to help you understand what's going on. Besides, you know the *rough* time interval that each script gets ran, but *when exactly* are these jobs ran? Let's pick on the final job:

```
52 6 1 * * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
```

Open your preferred web browser, and pay visit to:

<https://crontab.guru>

In the rounded gray input box, enter the following:

```
52 6 1 * *
```

And immediately above it, `crontab.guru` interprets it and spits out:

```
"At 06:52 on day-of-month 1"
```

In plain English: this job gets run at 6:52 in the morning on the first day of each month. Okay, that's neat and all, but what about the remaining lines? The root portion of the job tells cron to run this job as the root user. The final portion of this job reads:

```
test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

This is actually a series of commands that essentially reads: "Check and see if the file `/usr/sbin/anacron` exists, and is executable. `anacron` is another job scheduling system. If that file exists and is executable (`test -x`), then assume `anacron` is installed, and is the preferred system for handling scheduled tasks. We don't want to risk running the same tasks twice if `anacron` is installed, so do not run this cron job if the test statement returns true.

Otherwise, change directories to the root (`/`) directory. If that's successful, run the `run-parts` command with the arguments `--report` and `/etc/cron.monthly`. This has the effect of running all of the scripts on `/etc/cron.monthly` as the root user on the first day of month at 6:52AM.

Let's say you've dropped your updater script into `/etc/cron.daily`, but you have no indication that the script is running every day like it should be. Let's assume you'd like to run your update job at 6:05AM daily. What does that look like?

```
05 6 * * *
```

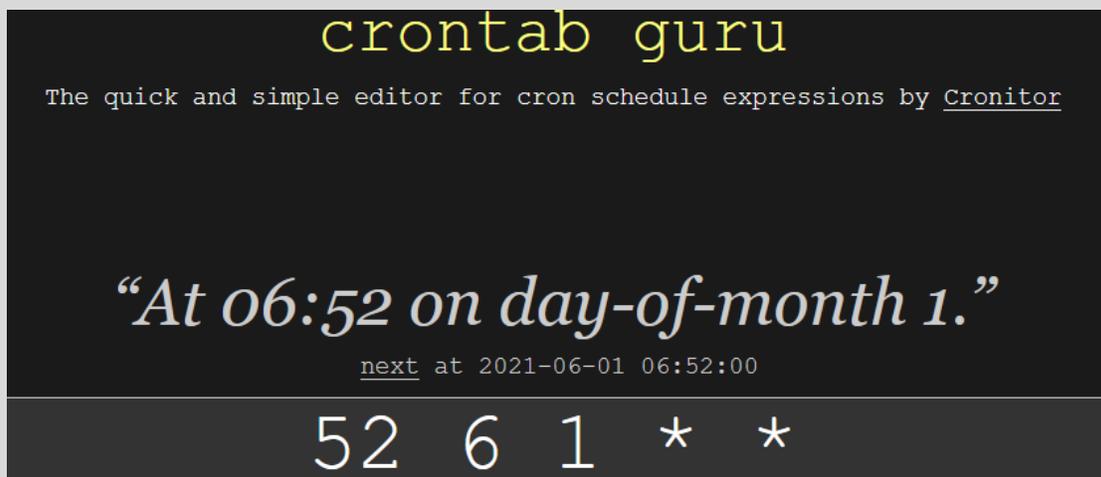
Plug that into `crontab.guru`, and you'll see that this translates to "At 6:05" (daily). I'm going to assume that the `updater` script is still located in both `/etc/cron.daily`, as well as `/home/[username]` (or root's `~/` directory). Run the following commands:

```
sudo su -
rm -rf /etc/cron.daily/updater
mkdir ~/adm_scripts/
chmod 700 ~/adm_scripts
cp /home/[username]/updater ~/adm_scripts/
chmod 700 ~/adm_scripts/updater
chown root:root ~/adm_scripts/updater
cp /etc/crontab /etc/crontab_bak
echo "6 05 * * * root /root/adm_scripts/updater" >> /etc/crontab
```

This chain of commands makes users become root (unless they're already the root user) Then we remove the updater script from /etc/cron.daily – substitute daily for weekly, monthly, etc. if necessary. Next, we create the directory, adm_scripts in root's home directory. The fully qualified path is: /root/adm_scripts. Then we use chmod 700 to restrict permissions to the adm_scripts directory to just the root user. Next, we copy the updater script from the regular user's home directory. If students downloaded the updater script as the root user, run cp ~/updater ~/adm_scripts instead. Next, we change the file permissions to ~/adm_scripts/updater to where only root can access the file, then ensure that ~/adm_scripts/updater is owned by the root user and group. Next, we use the cp command to create a backup of /etc/crontab before we try mucking around with it. That way, if we mess anything up, we can restore the original crontab by running cp /etc/crontab_bak /etc/crontab. Finally, the command:

```
echo "6 05 * * * root /root/adm_scripts/updater" >> /etc/crontab
```

appends a new cron job to the very end of the /etc/crontab file. We're telling the crontab to run the script /root/adm_scripts/updater as the root user daily at 6:05AM.



```
root@Kali:~# whoami
root
root@Kali:~# rm -rf /etc/cron.daily/updater
root@Kali:~# mkdir ~/adm_scripts
root@Kali:~# chmod 700 ~/adm_scripts/
root@Kali:~# cp ~/updater ~/adm_scripts/
root@Kali:~# chmod 700 ~/adm_scripts/updater
root@Kali:~# chown root:root ~/adm_scripts/updater
root@Kali:~# cp /etc/crontab /etc/crontab_bak
root@Kali:~# echo "6 05 * * * root /root/adm_scripts/updater" >> /etc/crontab
root@Kali:~# tail -1 /etc/crontab
6 05 * * * root /root/adm_scripts/updater
```

20-13: Using this string of commands, along with crontab.guru, students can define when and where they want to execute the updater script without having to deal with run-parts or the /etc/cron.* directories at all.

Note: Another alternative to anacron, cron and /etc/crontab, run-parts and the /etc/cron.* directories is systemd-timers. Up until mid-2021, I had never heard of systemd-timers, but like with everything else systemd related, it's a reinvention of old wheels that is allegedly improved over other job management services and daemons. I have zero experience with systemd-timers, but if students are so inclined, it may be an alternative worth investigating. Here are some resources students can use to read up about systemd-timers, if they're so inclined:

<https://opensource.com/article/20/7/systemd-timers>

<https://unix.stackexchange.com/questions/278564/cron-vs-systemd-timers>

20.3 Setting up ntpd on Linux lab VMs

Back in chapter 14, students configured the pfSense VM to act as an NTP server. In order to make use of that service, the Kali, SIEM and IPS virtual machines (and/or any other virtual machines students add to their lab) need to be configured to synchronize against it. Log in to the VM students wish to configure (and in the case of Kali, open the terminal application), Then run the following commands:

```
sudo su -
apt-get update
apt-get -y install ntp ntpdate
systemctl stop ntp
ntpdate [LAN or OPT1 IP address]
vi /etc/ntp.conf
[after saving ntp.conf]
systemctl start ntp
ntpq -pn
```

The first command, `sudo su -`, provides access to the root account to run the remaining commands. The next two commands will install the ntp server `ntpd`, and the `ntpdate` utility via the apt package manager. Afterwards, students run `systemctl stop ntp` in order to stop the NTP service, so that we can use the `ntpdate` command to force an immediate clock synchronization. The reason why there are brackets `[LAN or OPT1 IP address]`, is because VMs on the LAN/Management network will need to run `ntpdate 172.16.1.1`, while VMs on the OPT1/IPS 1 and 2 networks will need to run `ntpdate 172.16.2.1`. This corresponds to the NTP service on the *LAN* or *OPT1* IP address. After doing that, students will need to use `vi` or another text edit of their choice to modify the `/etc/ntp.conf` file. Remove any existing lines that start with the word "server", or "pool" followed by a hostname. These hostnames are usually NTP pools specific to that Linux distribution.

Replace them all with a single server line for the Management network (SIEM and IPS VMs):

```
server 172.16.1.1
```

Or the IPS network (e.g., the Kali VM):

```
server 172.16.1.2
```

And save the file. For students using alternative IP addresses and network configurations, substitute the IP addresses as necessary. Afterwards, run the command `systemctl start ntp`, followed by the command `ntpq -pn`. These two commands restart the NTP server, and then query the NTP server for statistics in order to confirm time synchronization is working correctly. Look at the field labeled *st*. This stands for stratum, and defines how many "layers" away from a clock/time source our system is.

NTP works sort of like a giant game of telephone. Certain systems on the internet are considered definitive time sources and are synchronized against atomic, radio or GPS-based clocks. The clock itself would be considered stratum 0, the direct time source. The server getting time data from the clock source is considered stratum 1. The next NTP server sync'ing against the server directly connected to the time source is considered stratum 2 and so on, and so forth. Each stratum introduces a little bit of delay and a slight loss of accuracy due to latency and other transmission effects. Without going too deep into the weeds, NTP has ways of compensating for those delays, but generally, the lower your stratum, the better. Most of the time, this number will be anywhere from 2 to 5. If this number is higher than that, or reads 16, that means the NTP source is really far away, or is not synchronizing. Check out the sidebar discussion below, *Out of Time*, for more information on troubleshooting NTP issues.

As students get more comfortable with their lab environment, additional VMs can request NTP from pfSense as well – just follow the same process as above (substituting the correct package manager commands and package names for `ntp` and `ntpdate` as needed), and remember to point Management network segment VMs to 172.16.1.1, and IPS1/2 VMs to 172.16.2.1 in the `/etc/ntp.conf` file. Once finished, remember to `exit` the root shell.

```

ayy@ips:~$ sudo su -
[sudo] password for ayy:
root@ips:~# apt-get update; apt-get -y install ntp ntpdate
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libopts25 sntp
Suggested packages:
  ntp-doc
The following packages will be REMOVED:
  systemd-timesyncd
The following NEW packages will be installed:
  libevent-core-2.1-7 libevent-pthreads-2.1-7 libopts25 ntp ntpdate sntp
root@ips:~# systemctl stop ntp
root@ips:~# ntpdate 172.16.1.1
11 May 20:02:06 ntpdate[475953]: adjust time server 172.16.1.1 offset 0.004370 sec
root@ips:~# vi /etc/ntp.conf
# Specify one or more NTP servers.

# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
pool 0.ubuntu.pool.ntp.org iburst
pool 1.ubuntu.pool.ntp.org iburst
pool 2.ubuntu.pool.ntp.org iburst
pool 3.ubuntu.pool.ntp.org iburst

# Use Ubuntu's ntp server as a fallback.
pool ntp.ubuntu.com

```

```

# Specify one or more NTP servers.
server 172.16.1.1

:wq!
root@ips:~# systemctl start ntp
root@ips:~# ntpq -pn
=====
remote           refid           st t when poll reach  delay  offset  jitter
=====
172.16.1.1       163.237.218.19 2 u  11  64   1   0.363 -1.508  0.000
=====

```

20-14: As the root user, students will need to install the ntp and ntpdate packages on the SIEM, IPS and Kali VMs via the apt package manager. Afterwards, use `systemctl stop ntp` to stop the ntp service, followed by `ntpdate 172.16.1.1` (172.16.2.1 for the Kali VM) to sync to the pfSense VM's NTP server. Next, students will need to modify the file `/etc/ntp.conf`. Remove all of the default NTP servers/pools configured in the file and replace them with the line `server 172.16.1.1` (172.16.2.1 for Kali). Afterwards use `systemctl start ntp` to restart the ntp server, and `ntpq -pn` to verify ntp is operating correctly.

Out of Time: Troubleshooting NTP Problems

Here are some things to consider if your VMs are having problems getting time from the pfSense NTP server (e.g., `ntpd` isn't working, or `ntpq -pn` is giving you stratum 16):

Chapter 14, section 14.4 Discusses the pfSense firewall policy. Make sure there is a firewall rule to allow access to the NTP server on 172.16.1.1 (LAN/Management) and/or 172.16.2.1 (OPT1/IPS1, 2). Remember firewalls process rules from top to bottom, and reject/drop rules take priority over allow rules.

Speaking of firewall rules, make sure port 123/udp is allowed outbound on your physical network and if not, ask your network admins what time/NTP servers should be used, instead.

Check for typos on the *Services > NTP* page in pfSense and/or `ntp.conf` on your VMs. Remember that incorrect data in will never result in correct data out.

Remember to point VMs in the IPS 1 or IPS 2 networks to the *OPT1* interface (172.16.2.1), and Management network VMs to the *LAN* interface (172.16.1.1)

If you are having DNS problems, that means you are having NTP problems. While your virtual machines sync to an IP address and do not require hostname resolution, the pfSense VM will need to be able to resolve hostnames to get time data from the NTP pool, if you're not using a private NTP server.

<(^_^)> my job
is done here!

(>'')> but you
didn't do any
thing

<(' - ' <)> that's
the best part