CISCO

Zero Trust in Resilient Cloud and Network Architectures

Early Release

RAW & UNEDITED

DHRUMIL PRAJAPATI, CCIE® NO. 28071

JOSH HALLEY, CCIE® NO. 11924

ARIEL LEZA

VINAY SAINI, CCIE® NO. 38448

ciscopress.com

CISCO

Zero Trust in Resilient Cloud and Network Architectures

Early Release

RAW & UNEDITED

DHRUMIL PRAJAPATI, CCIE® NO. 28071
JOSH HALLEY, CCIE® NO. 11924
ARIEL LEZA
VINAY SAINI, CCIE® NO. 38448

ciscopress.com

Zero Trust in Resilient Cloud and Network Architectures

Josh Halley, CCIEx3 No. 11924
Dhrumil Prajapati, CCIEx2 No. 28071, CCDE No. 20210002,
Ariel Leza,
Vinay Saini, CCIE No. 38448, CWNE No. 69, CCDE No. 20240032

A NOTE FOR EARLY RELEASE READERS

With Early Release eBooks, you get books in their earliest form-the author's raw and unedited content as they write-so you can take advantage of these technologies long before the official release of these titles.

Please note that the GitHub repo will be made active closer to publication.

If you have comments about how we might improve the content and/or examples in this book, or if you notice missing material within this title, please reach out to Pearson at PearsonITAcademics@pearson.com

Zero Trust in Resilient Cloud and Network Architectures

Josh Halley, Dhrumil Prajapati, Ariel Leza, Vinay Saini

Published by:
Cisco Press
Hoboken, New Jersey

No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Please contact us with concerns about any potential bias at www.pearson.com/en-us/report-bias.xhtml

$PrintCode

Library of Congress Control Number: 2025934165

ISBN-13: 978-0-13820-460-0

ISBN-10: 0-13-820460-8

Warning and Disclaimer

This book is designed to provide information about segmentation concepts, network access control, and resilient cloud and enterprise network architectures. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the authors and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers’ feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Please contact us with concerns about any potential bias at https://www.pearson.com/report-bias.xhtml.

Head of IT & Professional Learning, Enterprise Learning and Skills
Julie Phifer

Alliances Manager, Cisco Press
Caroline Antonio

Executive Editor
James Manly

Managing Editor
Sandra Schroeder

Development Editor
Ellie C. Bru

Senior Project Editor
Mandie Frank

Copy Editor
Chuck Hutchinson

Technical Editors
Asier Arlegui Lacunza
Istvan Matyasovszki

Designer
Chuti Prasertsith

Composition
codeMantra

Indexer

Proofreader

About the Authors

Josh Halley is a Principal Architect in the office of the CTO at Cisco Systems, where his current role focuses on domains related to up-and-coming and burgeoning new technology trends and innovation, engaging in large and complex strategic negotiations, and working with C-Levels to set technology strategy and direction within their organizations. Over the years, Josh has worked in many differing roles at organizations ranging from technology companies to banking and finance and management consulting.

Technology has long been a passion and key area of interest throughout his career, leading Josh to pursue three CCIE certifications to advance his expertise across multiple domains. During his years at Cisco, he has been engaged in technology projects and pre-sales activities all around the world, which has given him the opportunity to learn different cultures, perspectives, and customs.

Within Cisco, Josh has worked side by side with many of the company's business units, product managers, and technical marketing engineers and has been directly involved in the creation and deployment of over 180 new software features across multiple technology domains, many of which are associated with zero trust capabilities seen in Cisco's products and portfolio today.

Today Josh maintains his focus on technology, driving innovation with customers in the field and actively participating in the creation of patents and white papers and new product and service offerings for Cisco. Further to his work within Cisco, Josh is also an open-source software advocate who actively participates in working groups from the Cloud Native Computing Foundation, providing him an opportunity to share his knowledge and expertise with the wider open-source community.

Dhrumil Prajapati is a Principal Architect within Cisco CX's GES Architectures team. His focus lies on designing multidomain networks, and he has been offering a complete lifecycle of professional services and architecture advice for the past 15 years. His expertise extends to serving enterprise, government, and service provider entities across the globe. His services are designed to assist clients in planning, designing, deploying, managing, and interoperating all networking technology domains within their private or public infrastructure and application environments.

In his networking career, Dhrumil has designed networks for over 175 organizations, which inspired him to write a book on the subject. He is a coauthor of *Designing Real-World Multi-Domain Networks* (https://www.ciscopress.com/store/designing-real-world-multi-domain-networks-9780138037215) and *Cisco SD-Access for Industry Verticals* (https://cs.co/sda-verticals-book). He also holds patents and has given multiple presentations in Cisco Live on SD-Access, multidomain, and automation.

Dhrumil holds dual CCIEs in Enterprise Infrastructure and Service Provider, as well as a CCDE, in addition to other leading technical certificates. He also assists the Cisco Certifications team by reviewing and providing feedback for Cisco certificate. In addition, he leads several initiatives within Cisco CX aimed at driving delivery standardization and enhancing efficiency through automation innovation.

Currently residing in Apex, North Carolina, Dhrumil has a passion for motor racing, woodworking, and innovative electronics that enhance human life. His wife, Devanshi, and son, Ram, bring pure joy to his life, adding a touch of fun every day.

Ariel Leza, an ardent entrepreneur and technology aficionado, currently excels as a Cloud Solutions Architect in the EMEA CX CTO office at Cisco. His expertise is centered on guiding clients toward achieving their unique objectives through intricate cloud and IT architectures. Ariel's journey in the IT realm commenced during his military service as a network engineer in 2011, uncovering his passion for networking and cybersecurity. Today, he stands as a results-driven entrepreneur with a proven track record in technical sales and product positioning, fueled by enthusiasm for creating new businesses.

With over a decade of experience in the field, Ariel specializes in cloud and cross-domain IT architectures, focusing on cloud-native technologies, open-source software, networking, virtualization/compute, and data center software stack components. As a technology leader and technical solution architect in the Israeli market, he has demonstrated an exceptional ability to convey complex technological concepts in simple terms, tailoring his message to diverse audiences and emphasizing Cisco's unique value proposition.

In his role under the Customer Experience CTO Office of EMEA, Ariel has been instrumental in driving strategic programs and building strategic relationships with key accounts, leading to significant achievements such as engagement with top EMEA accounts, patent submissions, and pioneering deployments in blockchain technology and cloud-native services. His skills in cloud-native platforms, DevOps, Linux, and cloud computing are complemented by his deep understanding of decentralized applications, cryptocurrency, microservices, and Web3 blockchain architectures.

Outside of his professional endeavors, Ariel is a passionate participant in open-source and community projects. He also dedicates his time to global business ventures and the management of his demo Innovation Lab initiative. Ariel's commitment to making a meaningful impact as an entrepreneur and businessperson is driven by his belief in taking active responsibility for our civilization's actions and consequences.

With a vibrant spirit and an undying passion for technology and innovation, Ariel Leza continues to make significant strides in the tech world, inspiring others and driving successful customer outcomes with his dynamic approach and creative solutions.

Vinay Saini is a technologist, inventor, author, and mentor with over two decades of experience in the networking industry. As a Principal Architect at Cisco Systems (Customer Experience Group), Vinay guides customers across diverse verticals—including enterprise and IIoT—on their digital transformation journeys.

Vinay holds a bachelor of technology degree in IT and an MBA in international business. He is the first individual in India to achieve dual expert-level certifications: CWNE (#69) and CCIE (#38448). Additionally, Vinay is CCDE certified (#20240032) and holds many other industry credentials. He also actively contributes to Cisco Certification programs.

With a portfolio of over 100 patents filed and numerous defensive innovations, Vinay is at the forefront of technological innovation. As a sought-after speaker at events like Cisco Live, Vinay is also passionate about mentoring professionals worldwide, fostering their growth in technical and behavioral domains to help them reach their full potential.

About the Technical Reviewers

Asier Arlegui Lacunza, CCIE No. 5921, has been with Cisco since 1998 and currently works as a Principal Architect in Cisco’s Customer Experience organization. In the past 20+ years of his career at Cisco, he has worked as a technical architect on a wide range of enterprise (data center, campus, and enterprise WAN) and service provider (access and core networking) technology projects, with a focus on network automation. He holds a master’s degree in telecommunications engineering from Public University of Navarre, Spain.

Istvan Matyasovszki has been with Cisco since 2007 and is currently a Security Solutions Architect in Cisco’s EMEA Customer Experience organization, with a special interest and focus on network access control, remote access, intrusion detection, and microsegmentation. Istvan holds a PhD in computer engineering from the University of Limerick in Ireland and, before that, worked as a UNIX/Linux system administrator for several years.

Dedications

Josh:

I would like to dedicate this book to my two bloodhounds. Thanks for keeping me company and spending many a late night and early morning with me while I was authoring my chapters.

Dhrumil:

Publishing a first book is a core memory, and the third one sets an example for many. I want to dedicate this book to my wife, Devanshi, who has given unconditional love and been a true partner as I achieve my goal. It goes to the silent sacrifices you have made that brought me to where I am today.

My son, Ram, I never imagined I would write a book, but your curiosity in reading books from a small age and gaining as much knowledge as possible has inspired me to write my third book.

Ariel:

I would like to dedicate this book to the open-source community, including the developers and advocates championing the progression of the next generation of decentralized technologies.

Vinay:

To my parents, for instilling in me the values of perseverance and curiosity.

To my wife, Sowbhagya, for her unwavering support and patience throughout this journey.

To my children, Vihaan and Pranav, for inspiring me to think about a better future.

To my colleagues, friends, and mentors, whose insights, encouragement, and collaboration have been invaluable.

This book is dedicated to you.

Acknowledgments

Josh:

There were so many talented individuals who supported me on the journey of writing this publication that I feel it could fill an entire new chapter alone. With that clearly not being possible, I would like to begin by sending my sincere thanks to our reviewers, Asier and Istvan, who patiently provided us with feedback, comments, and creative criticism. Thanks, guys, for taking on this on top of your already hectic and busy day jobs to support me. Thanks also to our publisher, Pearson, and its amazing team who put up with our challenging schedules, updates, changes, and erroneous grammar. I would like to share a very special thank you to Meg Rainbow for her contribution on Chapter 1, Lee Sudduth for his contribution on Chapter 14, and Marcin Hamroz and Jaroslaw Gawron for their contributions on Chapter 22; these contributions added real-world perspectives from other vantage points that certainly enriched the overall manuscript. THANK YOU! Additionally, I would like to thank Kevin Regan, Alex Burger, Gino Corleto, Keith Baldwin, Mahesh Nagireddy, Jerome Dolphin, Sandeep Joseph, Einar Nilsen-Nygaard, Jonothan Eaves, Jeremy Cohoe, Kevin van Hengel, and Darrin Miller for their help and guidance along the way. To my leadership team (past and present), Adele Trombetta, Michael Kaemper, Markus Gierlich, and Haim Pinto, thank you for being supportive of my taking on this endeavor. And, last but certainly not least, I would like to thank my loving wife and children for their all-enduring support and motivation.

Dhrumil:

I would like to thank my leaders—Larry Hohmann, Mike Shomaker, and Jason Penn—for supporting me through this journey.

My peers, colleagues, and esteemed architects who have always shown their technical aptitude and willingness to help.

Lastly, to Amit Singh, without whom Chapter 12 of this book would not have been complete!

Ariel:

I would like to thank the great team at Pearson for their support in reviewing my content and supporting me along the way. I would also like to share a special thank you to my family and fiancé, Lilian, for their patience and support through the process of creating this book.

Vinay:

I would like to express my heartfelt gratitude to my management at Cisco for their unwavering support and encouragement, constantly motivating me to go beyond my defined role and explore new possibilities. Their guidance has been instrumental in shaping my professional growth.

I would also like to extend my appreciation to my colleagues, mentors, and the broader Cisco community for their collaboration, insights, and shared passion for innovation. Their support and camaraderie have made this journey truly enriching.

This book is a reflection of the collective inspiration and knowledge I have gained from working alongside such incredible individuals. Thank you for being a part of my journey.

Contents at a Glance

Part 4: Cloud Security

Part 5: Key Customer Case Studies

Part 6: Integrations and Automation

Reader Services

Register your copy at www.ciscopress.com/title/ISBN for convenient access to downloads, updates, and corrections as they become available. To start the registration process, go to www.ciscopress.com/register and log in or create an account*. Enter the product ISBN 9780138204600 and click Submit. When the process is complete, you will find any available bonus content under Registered Products.

*Be sure to check the box that you would like to hear from us to receive exclusive discounts on future editions of this product.

Contents

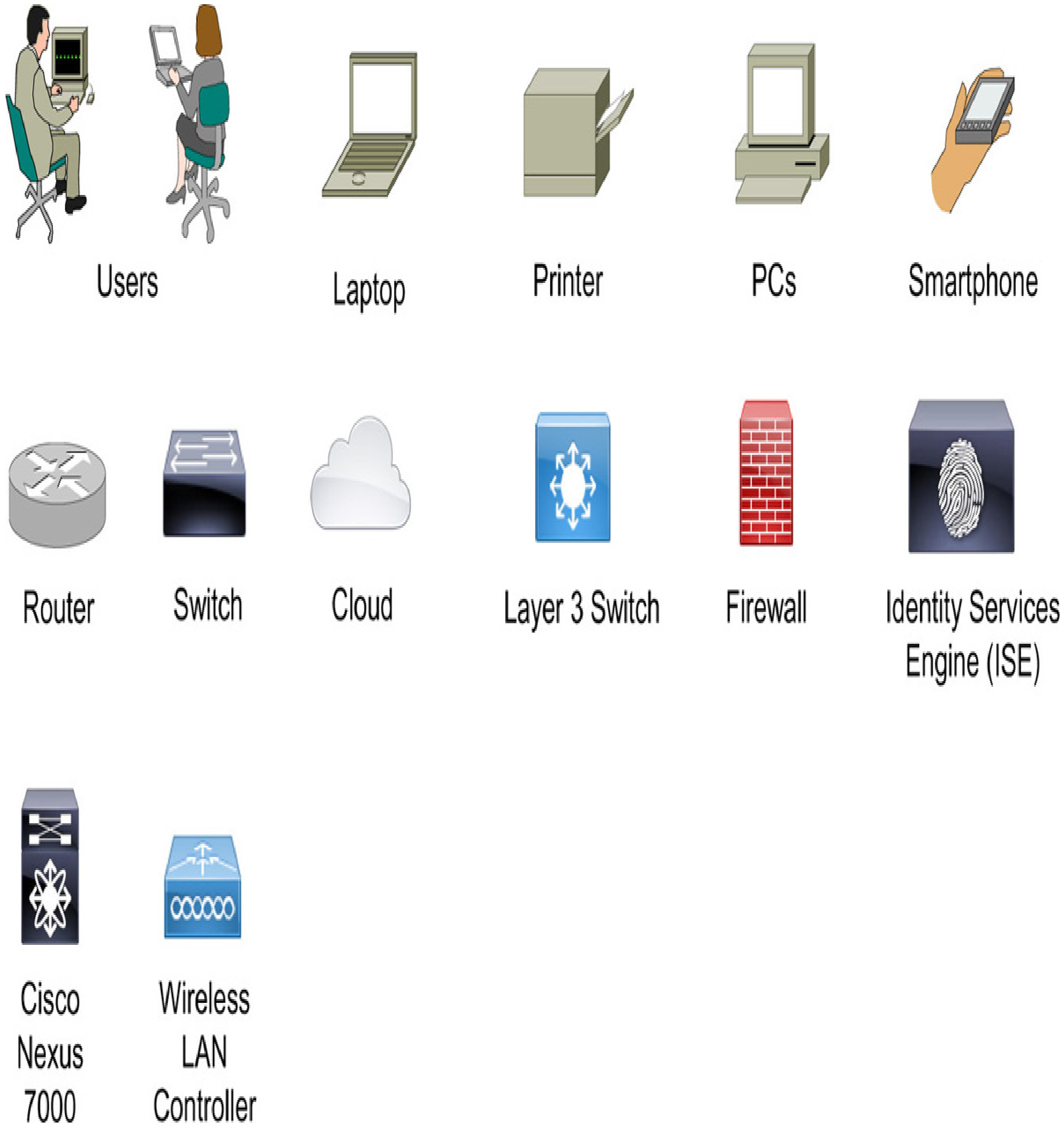
Part 2: Network Automation Capabilities in Software Defined Architectures

Part 3: Deployment Best Practices, Architectures, and Use Cases

Part 4: Cloud Security

Part 5: Key Customer Case Studies

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Introduction

The idea for this book originally came into existence during a time that the four authors were heavily involved in global deployments of large and complex networks focused on zero trust architectures. Pre-COVID, many of the authors were traveling up to 42 weeks a year, meeting with customers, supporting large deployments onsite, and hearing first-hand about the challenges, pitfalls, and gotchas they were experiencing from deploying advanced and new technology suites that could lead to a more robust and secure network architecture.

Over time, the team observed the technology mature, the escalations and teething issues seen in the early days of deployments of zero trust networks subside, and customers shift gears to realize their visions of a more automated, dynamic, and secure estate, focusing on the principles of providing only the requisite access for a given service. In addition to the maturity of the technology, the expansion of domains to adopt zero trust principles moved beyond the campus, to other domains and verticals, such as edge compute, the data center, cloud, and deployment in containerized form factors in AI agents. These adoptions further cemented the need to ensure a robust and secure estate, regardless of the location of the user, endpoint, workload, or autonomous/semi-autonomous software process.

The key focus areas of this book are not specific to a single technology; this title provides a focus on the domains of resilience, automation, security, and cloud architectures. While separate in their own right, when utilized together, they can result in a scalable, secure, and future-proof deployment.

Who Should Read This Book?

This book is aimed at security engineers, security analysts, site reliability engineers, network engineers, architects, and operations teams who have a key focus on optimizing their network deployments, increasing security, resilience, and automation within their organization. Given the experience of the authors, a heavy focus has been placed on real-life examples from their years of deployment experience globally. While it would have been easy to describe the solutions through rose-colored lenses, this book provides a candid overview of the good, the bad, and the ugly that you can experience on your journey toward building an automated, resilient, and secure enterprise estate.

How This Book Is Organized

Chapter 1, "Zero Trust Demystified?": This chapter provides an overview of the importance of zero trust in an organization's network.

Chapter 2, "Secure Automation and Orchestration Overview": This chapter provides an overview of the automation and orchestration strategies available during the writing of this book.

Chapter 3, "Zero Trust Network Deployment": This chapter provides an overview of zero trust deployment on secure service edge and other architectures.

Chapter 4, "Security and Segmentation": This chapter takes you on a deep dive into micro- and macrosegmentation concepts and explains where they can be best applied for a modern enterprise network.

Chapter 5, "DHCP and Dynamic Addressing Concepts": This chapter provides an overview of dynamic addressing and security concepts related to its use.

Chapter 6, "Automating the Campus": This chapter covers concepts of campus automation and how to achieve them securely.

Chapter 7, "Plug-and-Play and Zero-Touch Provisioning": This chapter describes aspects of secure zero-touch and plug-and-play onboarding of network devices in enterprise domains, including resilience automation aspects.

Chapter 8, "Routing and Traffic Engineering": This chapter covers concepts of routing and traffic engineering, showing how proper planning and foresight can lead to building the most resilient networks.

Chapter 9, "Authentication and Authorization": This chapter details concepts associated with authentication and dynamic authorization in enterprise architectures.

Chapter 10, "Quantum Security": This chapter explains the deployment of quantum-based security and explores the concepts associated with how quantum security is relevant to a network estate.

Chapter 11, "Network Convergence and Considerations": This chapter explores concepts of convergence in software-defined architectures for enterprise, data center, and security domains in detail.

Chapter 12, "Software-Defined Network Deployment Best Practices": This chapter covers the best practices of SDN networks and how to deploy them.

Chapter 13, "Wired and Wireless Assurance": This chapter explores enterprise wired and wireless deployment constructs from a resilience perspective, including security options for guest network access in Cisco cloud-based Meraki and on-premises deployments.

Chapter 14, "Large-Scale Global Software-Defined Network Deployment Best Practices": This chapter focuses on a global Meraki network deployment, whereby Infrastructure as Code concepts and approaches are used to simplify and scale the deployment.

Chapter 15, "Cloud-Native Security Foundations": This chapter covers key security principles for cloud-native environments, focusing on Zero Trust, workload protection, IAM, encryption, and compliance. It also explores CSPM, infrastructure as code (IaC) security, and AI-driven threat detection.

Chapter 16, "Cloud-Native Application Security": Focusing on securing applications in dynamic environments, this chapter covers DevSecOps, CI/CD security, API protection, and OWASP best practices. It also explores Web3-based identity (DID) and AI-driven security automation.

Chapter 17, "Data Center Segmentation On-Prem to the Cloud": This chapter examines segmentation strategies for hybrid and multi-cloud environments, highlighting Zero Trust, microsegmentation, and policy enforcement. It also explores cloud-native segmentation models and blockchain-based security.

Chapter 18, **"Using Common Policy to Enforce Security":** This chapter discusses unified security policies, IAM best practices, and automated enforcement using CASBs and SOAR. It also covers software security frameworks (SDLC to SSDLC) and OWASP SAMM for security maturity.

Chapter 19, **"Workload Mobility: On-Prem to Cloud":**

This chapter explores workload migration strategies, Zero Trust integration, data security, and compliance. It also covers post-migration optimization, including cost management and observability tools.

Chapter 20, **"Resilience and Survivability":** The chapter focuses on key concepts around redundancy, survivability, and resilience for network architectures in the context of routed network deployments and security architectures.

Chapter 21, **"Zero Trust in Industrial Manufacturing Vertical":** This chapter explores OT/IoT-type deployments in the domain of security and zero trust.

Chapter 22, **"Third-Party SDN Integrations":** This chapter explores the integration of third-party devices and systems into a broader zero trust architecture and domain.

Chapter 23, **"Infrastructure as Code":** This chapter describes key concepts associated with deploying automation for Infrastructure as Code in secure enterprise networks.

Part 1: Zero Trust Fundamentals

Chapter 1. Zero Trust Demystified

In this chapter, you will learn about the following:

- Zero trust
- How it all began
- Security standards
- People, processes, and technology
- On-premises vs. cloud
- Hybrid environment recommendations
- Security certification (FIPS and others)

Definition of Zero Trust

Zero trust is a security concept and framework that fundamentally challenges the traditional notion of network security, which assumes that everything within an organization's network perimeter can be trusted. Instead, zero trust operates on the principle of "never trust, always verify" and implies that no user or device, whether inside or outside the network, should be automatically trusted. It requires strict identity verification and access controls for every user and device attempting to access resources, and it employs continuous monitoring to detect and respond to potential threats. A common misconception about zero trust is that it is a single product or technology that can be installed to achieve complete security when, in fact, it is a comprehensive approach involving multiple technologies and strategies, including identity management, endpoint security, and network segmentation. Additionally, zero trust is often mistakenly thought to imply a lack of trust in employees, whereas it is actually about enhancing security through rigorous verification processes. Misuse of the term can occur when organizations claim to have a *zero trust architecture (ZTA)* without implementing the fundamental continuous monitoring and validation processes that are core to the zero trust philosophy.

How It All Began

Around 2004, the Jericho Forum, an international group of IT security experts, started to question traditional perimeter security approaches and promoted the concept of *de-perimeterization*, which suggested that security should be built around assets and data rather than a network perimeter. While these experts did not yet use the term *zero trust*, the principles for which the Jericho Forum advocated laid the groundwork for what would later become zero trust.

Although the term *zero trust* emerged as early as 1994, it is generally accepted that the concept of zero trust was introduced in 2009–2010 by John Kindervag, a principal analyst at Forrester Research. Kindervag realized that traditional perimeter-based security models, which assumed that everything inside the network was safe, were no longer effective. In *No More Chewy Centers: Introducing the Zero Trust Model of Information Security*, Kindervag challenged the broadly applied concept of "trust but verify." Instead, he proposed a model based on the principle of "never trust, always verify." Kindervag argued that trust is a vulnerability and that organizations should assume that threats could come from both inside and outside the network. The notion of trust is a human emotion that had been misused in computer networks for no technical reason because it is not measurable and therefore should not be applied. The increasing complexity of networks, along with the rise of cloud computing, mobile devices, and remote work, made it clear that security needed to shift toward protecting individual assets rather than just focus on the perimeter. The perimeter-focused security model (a walled castle)—where once you were in, you were “trusted”—no longer served its purpose due to this complexity. A new approach was needed where security measures are implemented networkwide, requiring verification for every access request, regardless of its origin.

One of the most compelling real-world implementations of zero trust principles is Google’s BeyondCorp initiative. This initiative was born out of Google’s realization that perimeter security was no longer adequate. The zero trust model was launched in 2014, following the Operation Aurora attack in 2009, where Google and several other companies were targeted by sophisticated cyberattacks. BeyondCorp aims to eliminate the traditional security perimeter, treats all network traffic as untrusted, and focuses on securing access to internal applications without relying on virtual private networks (VPNs) or traditional perimeter defenses. Google shifted to a model where access is granted based on device health, user identity, and the context of the access request, rather than merely the network location. It emphasized identity and device security, combined with continuous verification.

This transition was not without its challenges. Google’s engineers had to overhaul existing security policies, develop new tools, and ensure that the new model did not disrupt daily operations. However, the effort paid off. BeyondCorp has enabled Google to offer its employees seamless and secure access to corporate resources from any location, thereby enhancing both security and productivity. The success of BeyondCorp has inspired numerous organizations to explore and adopt zero trust principles.

Following Google’s example, many companies started developing their own zero trust models. Major tech companies and cybersecurity vendors began to align their products and solutions with zero trust principles.

In 2010, the Institute for Security and Open Methodologies (ISECOM) played a pivotal role in the development and distribution of zero trust concepts by emphasizing the need for rigorous security methodologies that challenge traditional assumptions of trust within networks. ISECOM’s work highlighted the inherent vulnerabilities in perimeter-based security models and underscored the necessity of a more granular, verification-based approach to access control. This was a crucial contribution to the evolving security landscape, because it helped lay the groundwork for organizations to adopt strategies that assume breaches are inevitable and focus on minimizing trust zones and continuously verifying every request for access. By advocating for these principles, ISECOM significantly influenced the adoption and implementation of zero trust architectures across various industries. This research community dedicated a full chapter of its 2010 *Open Source Security Testing Methodology Manual (OSSTMM)* to trust and trust analysis. The work formulated this concept elegantly: “As part of OpSec, trust is one part of a target’s porosity. Where security is like a wall that separates threats from assets, trust is a hole in that wall. It is wherever the target accepts interaction from other targets within the scope.”

Figure 1-1 depicts the timeline of the important zero trust development milestones.

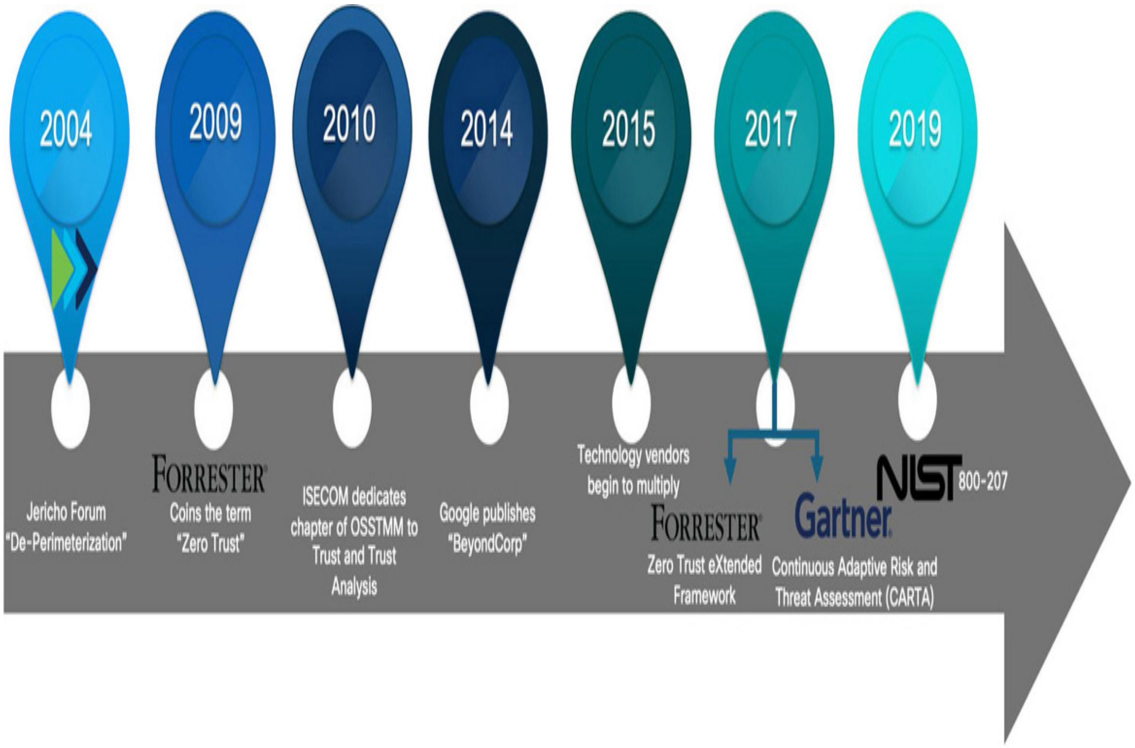


Figure 1-1 *Zero Trust Historical Timeline*

Why We Need Zero Trust

The need for zero trust has been underscored by the evolving threat landscape and the dissolution of the traditional network perimeter. The *evolving threat landscape* refers to the dynamic and continuously changing nature of cyber threats that organizations and individuals face. This landscape is influenced by the constant advancements in technology, which provide new tools for attackers and new vulnerabilities to exploit. As digital transformation accelerates with trends like cloud adoption, remote work, Internet of Things (IoT) devices, and artificial intelligence (AI), attack surfaces increase, giving cybercriminals more opportunities to breach systems. The threat landscape evolves not only in terms of volume but also in sophistication, as attackers use advanced techniques like social engineering, AI-powered malware, and highly targeted ransomware attacks to achieve their goals.

Additionally, the evolving threat landscape is shaped by emerging risks such as those posed by quantum computing, which could render current cryptographic standards ineffective, making sensitive data vulnerable to new types of breaches. Nation-state actors and organized cybercrime groups continuously refine their tactics to evade detection, aiming at critical infrastructure, financial systems, and personal data for strategic gain. Consequently, security strategies must also evolve to counter these threats, requiring approaches like zero trust that emphasize flexibility, continuous monitoring, and adaptive defenses.

Cyber threats have become more advanced, leveraging cutting-edge technologies such as artificial intelligence, machine learning (ML), and automation to create more effective and harder-to-detect attacks. Threat actors are exploiting a wider range of vulnerabilities, including those in cloud services, mobile devices, IoT devices, and remote work setups, increasing the potential points of entry for attacks. The frequency of cyberattacks is on the rise, with increasingly larger and more impactful breaches occurring. Ransomware, distributed denial-of-service (DDoS) attacks, and data breaches are becoming more common and disruptive. Attacks are often highly targeted, focusing on specific organizations or industries with valuable data or critical infrastructure. Advanced persistent threats (APTs) involve prolonged and targeted cyber espionage campaigns aimed at compromising sensitive information. Cyber criminals constantly adapt their strategies to bypass traditional security measures. This includes the use of social engineering, phishing, and spear-phishing attacks to deceive individuals into revealing confidential information or granting unauthorized access. Organizations must navigate an increasingly complex regulatory environment with stringent data protection and privacy laws, such as the General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA), which impose heavy penalties for noncompliance and data breaches.

With the rise of cloud computing, mobile workforces, and increasingly sophisticated cyber threats, traditional perimeter-based security models are no longer adequate to protect organizations. Insider threats, whether malicious or accidental, pose significant risks that conventional security measures often overlook. Modern IT environments are highly interconnected and complex, involving multiple vendors, third-party services, and cross-domain interactions. This complexity requires a more granular and dynamic approach to security: zero trust. Zero trust provides such a framework enabling organizations to implement stronger data protection, enhanced visibility, and more effective threat detection and response.

Organizations implement zero trust to reduce the risk of breaches, particularly in the era of cloud computing and remote work. A zero trust model mitigates the risks associated with users working from various locations and devices by ensuring that all access attempts are scrutinized. More precisely, zero trust is a cybersecurity framework that fundamentally challenges the traditional notion of network security, which typically relies on a strong perimeter defense. Unlike conventional models that operate on the premise of "trust but verify," zero trust adopts a "never trust, always verify" approach. This means that no entity—whether inside or outside the network—should be trusted by default. Instead, every access request must be authenticated, authorized, and continuously validated based on contextual factors such as user identity, device health, location, and behavior. At its core, zero trust emphasizes identity verification, least-privilege access, and continuous monitoring of every attempt to access network resources. Instead of assuming that a user or device inside the corporate firewall is safe, zero trust requires strict identity authentication and segmentation of network resources to limit access only to what is necessary.

Core Principles of Zero Trust

The principles of zero trust ensure a clear and unambiguous perspective on how to manage an architecture to promote security. The principles described here focus on core concepts that network and cloud architecture operators and security specialists should follow when designing, evaluating, and testing modern architectures.

Explicit Verification

Ongoing, persistent, and explicit verification of the security posture is a crucial aspect of achieving a viable zero trust deployment. The larger the number of data points that exist, the more flexible the ability to assess key and relevant criteria needed for higher or lower levels of authorization that would subsequently be granted to relevant network or application resources.

Some factors that are often utilized in performing these actions are user identity and their respective location, which could be a branch or head office, manufacturing line, remote access via VPN, or without, or even a triangulated location identified through cellular, wireless, or ambient sensor localization. In addition to the actual location, the endpoint and users themselves should be taken into close consideration and focus. Is the user a contractor or full-time employee? What has been their tenure in the organization? Have they had any discipline issues in the past that could lead to them being considered a higher risk or disgruntled? And have their patterns in system use and data access changed in recent weeks or months? Beyond looking at the identity of the users themselves, even if they have the best intentions to maintain valid security, what is the status of the endpoints that they are using? Do they have the latest patches installed? Are they up-to-date with the latest antimalware and antivirus software, and have they been used in areas that were considered insecure (open service set identifiers, or SSIDs) recently, which could have increased their exposure to a breach via lateral movement?

The technological capability to support with explicit verification in the context of zero trust is multifactor authentication (MFA). Having a single and simple password (as easy as it is for the user to remember) has never really been a very smart idea. Over time, the need to have more robust and complicated passwords, including mixed characters with digits and symbols with a minimum length, has become normal. However, generating hardened passwords alone is simply not enough to ensure that the credentials are secure enough to stem a network breach. For today’s application and network architectures, it has become normal to augment password authentication with further layers of security; the purpose is to increase the level of breach points that would be needed for a perpetrator to attempt to access network or system resources. Multifactor authentication can support users and systems in their means of authenticating with a higher degree of security.

Secondary or tertiary methods of authentication in this area could be through the use of biometrics, tokens, Short Message Service (SMS) or application push validation, or more advanced methods such as typing behavior using methods like keystroke dynamics.

Least-Privilege Access

Least-privilege access is the zero trust principle that ensures users, applications, and systems are granted only the minimum level of access necessary to perform their tasks or functions. This approach reduces the attack surface by limiting exposure to sensitive data and systems, minimizing the potential damage in case of a breach. By implementing controls such as role-based access control (RBAC), just-in-time (JIT) access, and granular permissions, least-privilege access prevents unauthorized or excessive access while maintaining operational efficiency and security.

Which access a given user is provided and what they are authorized to do are often limitations enforced via modern systems. The decision process used to identify what is accessed is often based on a role allocation. This role does not necessarily mean a one-to-one mapping of the user's role in the company, to their respective access rights, but under certain circumstances, there can be a vague indicator of baseline access provided for requisite systems needed for a given user's job and tasks. Role-based access comes down to the need to access a system to perform a task that can be limited down to the level of read-only, read-write, or a more explicit permission to access a given system. For instance, on Cisco devices, it is not uncommon to provide role-based access to permit the execution of one task with read-write access but limit other tasks to read-only or no access.

Does a user require continuous access to a given resource, or should that user's respective access be limited to only a given time, date, or maintenance window? This JIT access approach is becoming more common, particularly in environments that have critical systems. The question is: Just because a user is performing a task, should that user always be allowed to perform that task? Taking a relatively simple example, many users within an office environment are allowed to print, but should users be able to print documents to an office printer outside of the hours that the building is open and operational? Logically, having this access would not make a lot of sense, so why even permit the access for such a function? The same consideration could also be taken for network implementation teams that are responsible for configuring new network locations and configuring new features and functionalities. If these teams are only responsible for the implementation function but not necessarily involved in day 2 operations, should their access to configure critical systems still be allowed outside the hours of a maintenance window's completion time? *Just-enough access*, which in earlier systems was sometimes referred to as *lock-and-key access*, aims at addressing this problem space, limiting access to only what is required, and nothing more.

Assume Breach

The zero trust principle of *assume breach* operates on the premise that no system is ever fully secure and that a breach has either already occurred or will occur. This mindset shifts the focus from solely preventing attacks to proactively limiting their impact. It emphasizes implementing strong defenses, such as microsegmentation, continuous monitoring, and real-time threat detection, to contain threats and reduce their ability to move laterally within the network. By assuming breach, organizations are better prepared to identify, respond to, and mitigate attacks swiftly, enhancing overall resilience.

When operating under the assumption of a breach, organizations can implement various countermeasures to mitigate its impact. Two key strategies for addressing breaches are *breach containment* and *incident response and recovery*, both of which play a critical role in managing and resolving security incidents effectively.

Breach containment is the process of isolating a security threat to prevent it from spreading further within the network. It focuses on limiting the scope and impact of an attack by implementing measures such as microsegmentation, network isolation, and automated threat detection. Containment strategies also include restricting compromised accounts or devices and blocking malicious activities in real time. By quickly containing a breach, organizations can minimize damage, protect sensitive assets, and maintain operational continuity while preparing for further remediation efforts.

While the best-laid plans often aim to address every possible scenario and every possible outcome that could happen, sometimes unexpected things can still happen. For example, the threat of actors who are intent on accessing and breaching network systems for money, notoriety, revenge, or self-validation is an ongoing challenge. Even with the best systems in place, with the potential use of zero-day exploits, risk mitigation is sometimes reduced to risk minimization. For this reason, the focus of many security experts and plans within organizations needs to be around rapid identification and detection of the security incident at play, and the subsequent response of the threat, which may result in its respective containment and the remediation of system recovery, or getting the systems back in a working and functional state. This requires a good incident response and recovery process to be in place.

In addition to the core principles that were outlined previously, to ensure that a network can properly apply a viable zero trust deployment, it is important to consider that sometimes the biggest challenge in deployment of security is 10 inches away from the desk. To ensure that a well-structured and functional deployment can happen, the organization also needs to consider people and processes in detail.

Beyond the previously mentioned principles that require consideration when looking at deploying a zero trust architecture, key aspects are as follows (an illustration of these aspects is shown in Figure 1-2):

- **Segmentation:** The days of building a network or information system and putting all users and systems into one large subnet or all servers in a common segment without access control are long gone; this was never really a good practice. This lack of operational maturity manifested itself in the context of security incidents, a lack of auditability and a severe level of impact across key and critical systems within an IT system/network estate. In a shift away from earlier missteps in security, deploying segmentation has become a key aspect of modern-day security. Segmentation can be achieved in numerous ways—from the dedicated separation of network segments, application of per-system access rules, or dynamic allocation of policy based on system and identity criteria through software-defined network (SDN) technologies such as TrustSec. These capabilities are covered at length in Chapter 4, "Security and Segmentation," and other sections within this book, including best practices around their respective usage.

- **Identity and Access Management (IAM):** IAM is a framework of policies and technologies ensuring that the right individuals have appropriate access to technology resources in an organization. Deploying a structured, scalable, and explainable identity and access management solution is a key tenet in ensuring that rulesets, usage patterns, and access rights are appropriate to a particular user profile, role, and region that a user may be located in to spot unusual out-of-hour patterns of activity. Coupling a high level of confidence in the identity of a user or a system through tools such as identity federation, multifactor authentication, and role-based access control with network components and IT systems provides support for a more secure environment.

- **Network Access Control (NAC):** NAC is a security solution that manages and controls access to a network by enforcing policies, ensuring only authorized and compliant devices can connect. The use of network access control has become more pervasive over the years. This includes approaches to ratchet up network access by displacing legacy configurations such as port security that lacked centralized command and control and audit trail capabilities and awarding more advanced functionality through RADIUS and other protocols to ensure that only compliant users and devices can gain authorized access to operate within a given network domain, segment, or architecture.

- **Policy Enforcement Points (PEPs):** At which point of a network architecture is traffic carried until it is eventually dropped or permitted? Does it make sense to carry the traffic all the way over the wide area network (WAN) or up to the cloud to eventually discard it? Do observability or traffic pattern identification systems in the path need to be traversed prior to policy enforcement? Are polices crafted with pre– or post–Network Address Translation IP addressing? Does the audit trail for enforcement need to be central, or does it exist only on the enforcement point itself? Many of these questions come become clear when considering which points in the network are responsible for enforcement of policy. A proper grasp of which points to enforce is also key when heading into the realms of fault triage and troubleshooting.

- **Endpoint Security:** Client devices, often referred to supplicants or endpoints, are pervasive moving targets when it comes to security. With every software patch for the underlying operating system or agent that may be installed, further risks are exposed in terms of potential incompatibilities or gaps in security. By using endpoint detection and response systems and capabilities, customer estates can ensure that their respective policies for network security and compliance are adhered to. This goal can be achieved by validating and checking the state and versioning of software installed on the systems, including personal firewalls, antivirus and antimalware, and automation and monitoring agents.

- **Asset Management and Security Posture:** Asset management is a key tenet of maintaining a zero trust architecture. With counterfeit devices and assets appearing more prevalently than in the past, and ever-increasing complexity of modern supply chains, it is crucial to ensure that devices which were procured are indeed sourced from the vendor as expected. Assets need to be properly managed and tracked, allowing for the use of the asset database (a configuration management database [CMBD] or other) in a broader security context—from common failure trending to identification of insecure components that may have been included in a range of serial numbers within the corporate estate. Having control of this critical data helps security and network operation teams have the requisite tools to ensure that the right level of security is applied and enforced.

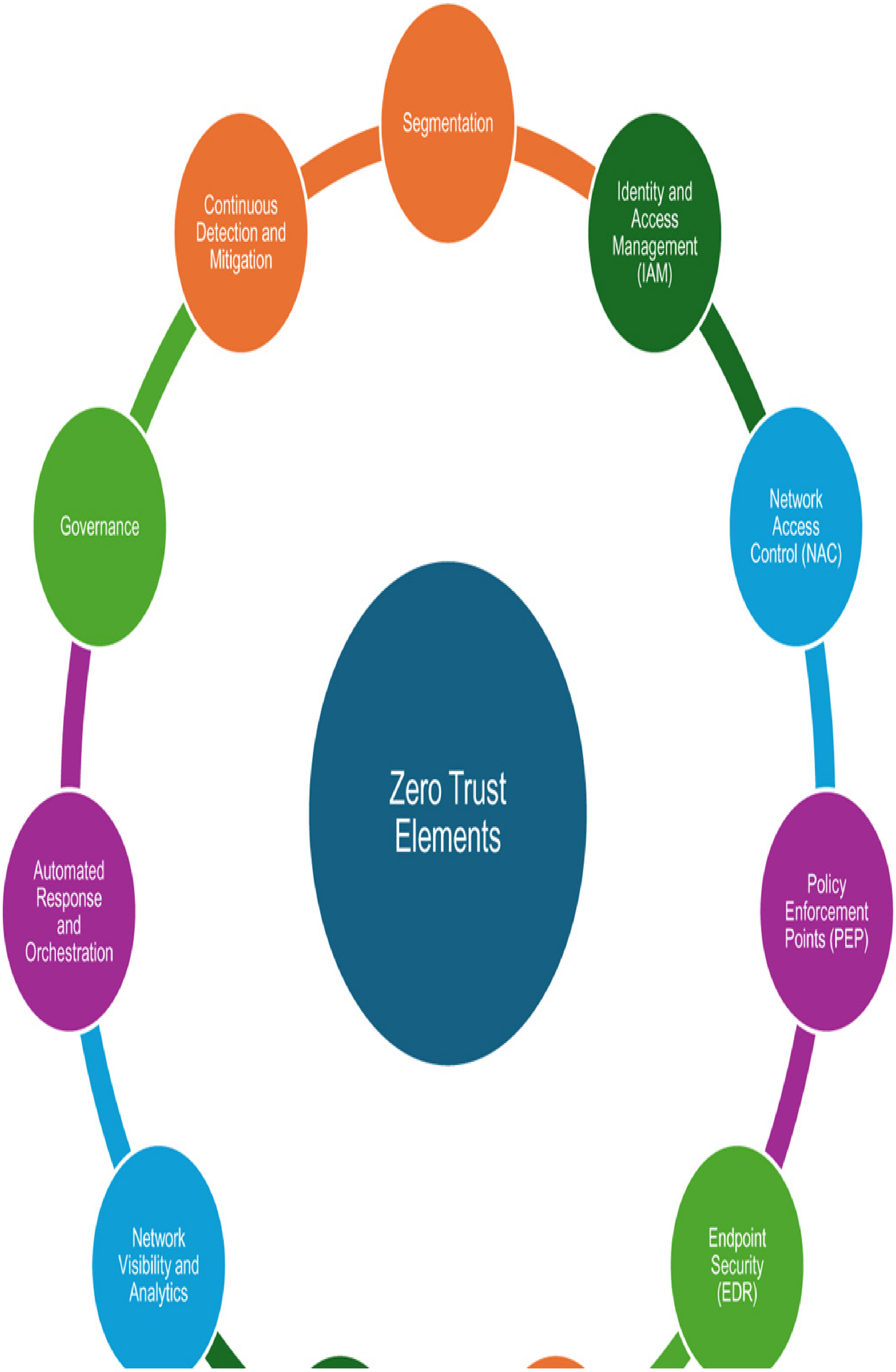
- **Encryption and Data Protection:** The way in which sensitive data is managed and encrypted both in transit and at rest is an important aspect in avoiding unauthorized access to the data. Furthermore, correctly classifying levels of the data helps ensure that the correct controls that support needed checks and balances are applied against the data, ensuring that sensitive data can be monitored and accessed only by parties who have the correct permissions. To improve the chances that sensitive information is not accessed by the wrong parties, data loss prevention (DLP) solutions allow for appropriate monitoring and improved protection.

- **Network Visibility and Analytics:** Understanding a normal and healthy baseline and state within a network allows appropriate analysis to take place against what is considered a "good" state within a given architecture or system. Leveraging tools that can analyze and visualize network traffic and the health of critical components that build up the resiliency of the solution can help in identifying anomalies and malicious behavior when looking at the deltas between the benchmarked "good" and/or "normal" state of systems, intermediate infrastructure components, and the flows of data that traverse them.

- **Automated Response and Orchestration:** Time is of the essence when attempting to reduce the harm and impact of a potential security incident or breach. Does the malicious party still have access to your architecture? Are remediation actions required on multiple systems? Which data has been compromised, and what impact does this have on the organization? Actions associated with response to an incident are key in minimizing the impact. Many such actions today represent automated executions based on the incident type or TrustScore.

- **Governance:** In today's organizations, people and processes need to be able to align to a common set of corporate security policies and guidelines. The purpose of these requirements is to ensure that the needs of the company can be enforced in the context of zero trust. The applied framework ensures that the necessary oversight is in place to allow for deployment, operation, and execution of policy that is flexible enough that it can be tailored to adjustments in the business's corporate strategy or change of vendor, products, or technical capabilities applied within the estate. Having the right governance structure in place allows the organization to rapidly pivot and address new and evolving threats, while maintaining a resilient and secure architecture.

- **Continuous Detection and Mitigation:** Security threats are not a single point-in-time activity that, once resolved and remediated, will never happen again. Ensuring that there is a constant ability to detect new threats, perform assessments and analysis of those threats, and apply the needed mitigation actions and strategy is key. With new exploits and threats appearing daily, ensuring that the needed rigor is applied to continuous detection is a key aspect of setting up a secure and relevant environment.


Segmentation
Identity and Access Management (IAM)
Network Access Control (NAC)
Policy Enforcement Points (PEP)
Endpoint Security (EDR)
Network Visibility and Analytics
Automated Response and Orchestration
Governance
Continuous Detection and Mitigation
Zero Trust Elements

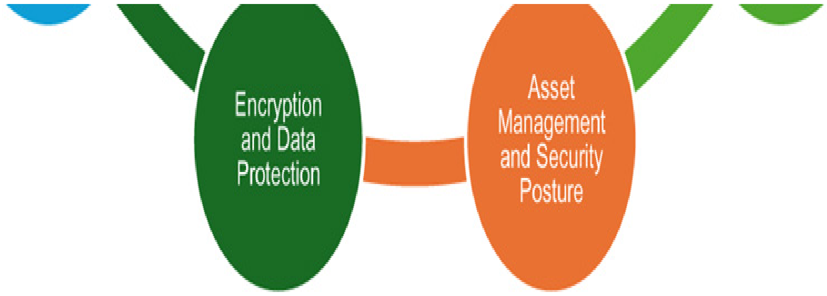

Encryption and Data Protection
Asset Management and Security Posture

Figure 1-2 *The Elements of Zero Trust*

Major Zero Trust Industry Standards

In 2018, the US National Institute of Standards and Technology (NIST) began working on a formal zero trust framework, recognizing its importance for both private and public sectors. The result was NIST Special Publication 800-207, released in 2020, which outlined the core principles of zero trust architecture and provided detailed guidance on implementing zero trust in complex environments (as shown in Figure 1-3). NIST's framework described zero trust as a shift from traditional perimeter-based security to a model focused on continuous verification, least-privilege access, and real-time threat monitoring. It emphasized the need for organizations to adopt security controls based on identity, device, and behavior, rather than network location.

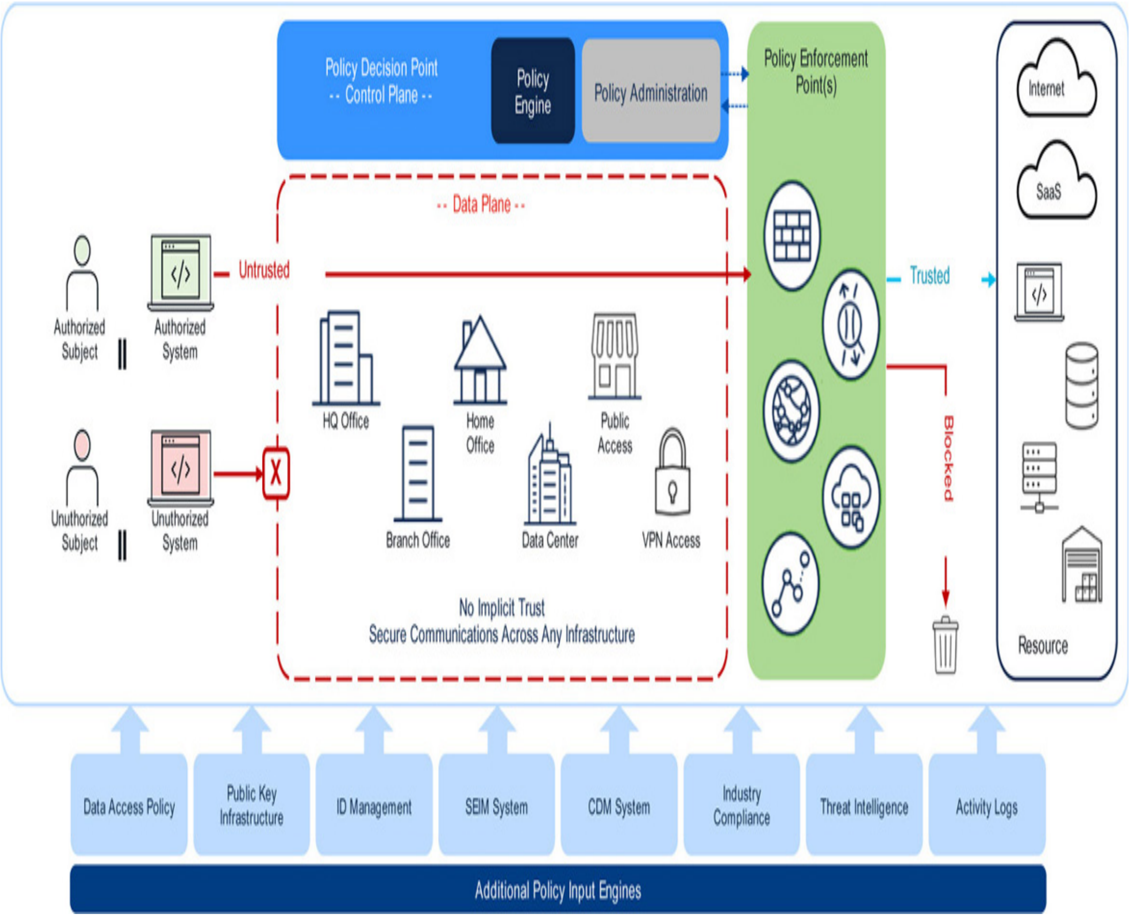


Figure 1-3 *Zero Trust Architecture Based on NIST 800-207 (Link: https://csrc.nist.gov/pubs/sp/800/207/final)*

The COVID-19 pandemic forced millions of employees to work remotely, greatly increasing the attack surface for cyber threats. The sudden need to support remote workforces accelerated the adoption of zero trust models, because traditional perimeter-based security approaches (such as VPNs) were unable to cope with the massive scale and complexity of remote access.

Organizations rapidly adopted zero trust network access (ZTNA) solutions, which granted users secure access to applications and data without relying on the network perimeter. Cloud adoption also surged, driving further investment in zero trust as it helped secure cloud environments by focusing on users, devices, and applications rather than networks.

In May 2021, following several high-profile attacks (e.g., SolarWinds), the US Executive Order on Improving the Nation's Cybersecurity explicitly mentioned zero trust, mandating federal agencies to develop plans for implementing zero trust architecture. This endorsement further validated the model and encouraged broader adoption across public and private sectors.

Several standards, guidelines, and best practices have been developed to help organizations implement zero trust principles effectively. While there is not a single, unified standard for zero trust, the following are key documents and frameworks that provide comprehensive guidance:

1. **NIST Special Publication 800-207**

   NIST SP 800-207 provides a comprehensive framework for implementing zero trust architecture (ZTA), emphasizing core principles such as continuous verification, least-privilege access, and network segmentation. The publication offers valuable guidance for organizations transitioning from traditional security models to zero trust by detailing key concepts like identity verification, which involves the ongoing authentication of both users and devices. It also highlights the importance of least-privilege access, ensuring that users are granted only the necessary permissions for specific tasks. Furthermore, the document discusses microsegmentation, which involves dividing networks into smaller zones to effectively contain potential threats, and continuous monitoring, which focuses on the ongoing evaluation of security posture, user behavior, and network traffic. Lastly, it underscores the significance of contextual access, which considers factors such as device health, user location, and the nature of the access request before granting permissions, thereby enhancing the overall security of the network.

2. **Forrester’s Zero Trust eXtended (ZTX) Framework**

   Forrester’s ZTX framework presents a strategic approach to zero trust by emphasizing a comprehensive focus on critical components such as data, workloads, people, devices, and networks. It serves as a practical guide for organizations aiming to assess their current security posture and effectively implement zero trust principles. By addressing these core areas, the ZTX framework helps organizations develop a more resilient security strategy that minimizes risk and enhances protection against modern cyber threats.

Link: https://www.forrester.com/blogs/category/zero-trust-security-framework-ztx/

3. **Gartner Continuous Adaptive Risk and Trust Assessment (CARTA)**

   The CARTA model is a strategic approach to cybersecurity that emphasizes the need for continuous, adaptive processes to manage risk and trust dynamically. Unlike traditional security models that rely on periodic assessments and static defenses, CARTA acknowledges the evolving nature of threats and the importance of real-time decision-making. It focuses on continuous assessment, real-time adaptation, and risk-based security decisions to keep pace with evolving threats and dynamic IT environments.

   Link: https://www.gartner.com/smarterwithgartner/the-gartner-it-security-approach-for-the-digital-age

4. **Cloud Security Alliance (CSA) Zero Trust Working Group**

   The CSA Zero Trust Working Group is dedicated to establishing best practices for implementing zero trust in cloud environments. This group focuses on producing valuable resources such as white papers, guidelines, and case studies that assist organizations in integrating zero trust principles within cloud and hybrid settings. By offering practical insights and detailed strategies, the working group plays a crucial role in helping organizations enhance their security frameworks and effectively address the unique challenges posed by cloud-based infrastructures.

   Link: https://cloudsecurityalliance.org/research/working-groups/zero-trust

5. **Cybersecurity and Infrastructure Security Agency (CISA) Zero Trust Maturity Model**

   The CISA developed a zero trust maturity model to guide organizations in transitioning from traditional security models to zero trust architectures by outlining distinct maturity stages (see Figure 1-4). This model is structured around key zero trust pillars, including identity, which involves continuous verification of users and devices; devices, which focus on monitoring and controlling device access; network, which emphasizes encrypting and monitoring all internal traffic; applications and workloads, which require securing and verifying applications; and data, which ensures data protection and restricted access. The maturity levels range from Traditional (Basic), characterized by legacy systems with minimal segmentation, to Optimal (High), where a fully implemented zero trust model features dynamic policy enforcement and continuous monitoring throughout the entire environment. Intermediate stages include Initial (Low), with basic policy implementation and access control, and Advanced (Medium), with sophisticated identity access controls and dynamic perimeter defenses.

Link: https://www.cisa.gov/zero-trust-maturity-model

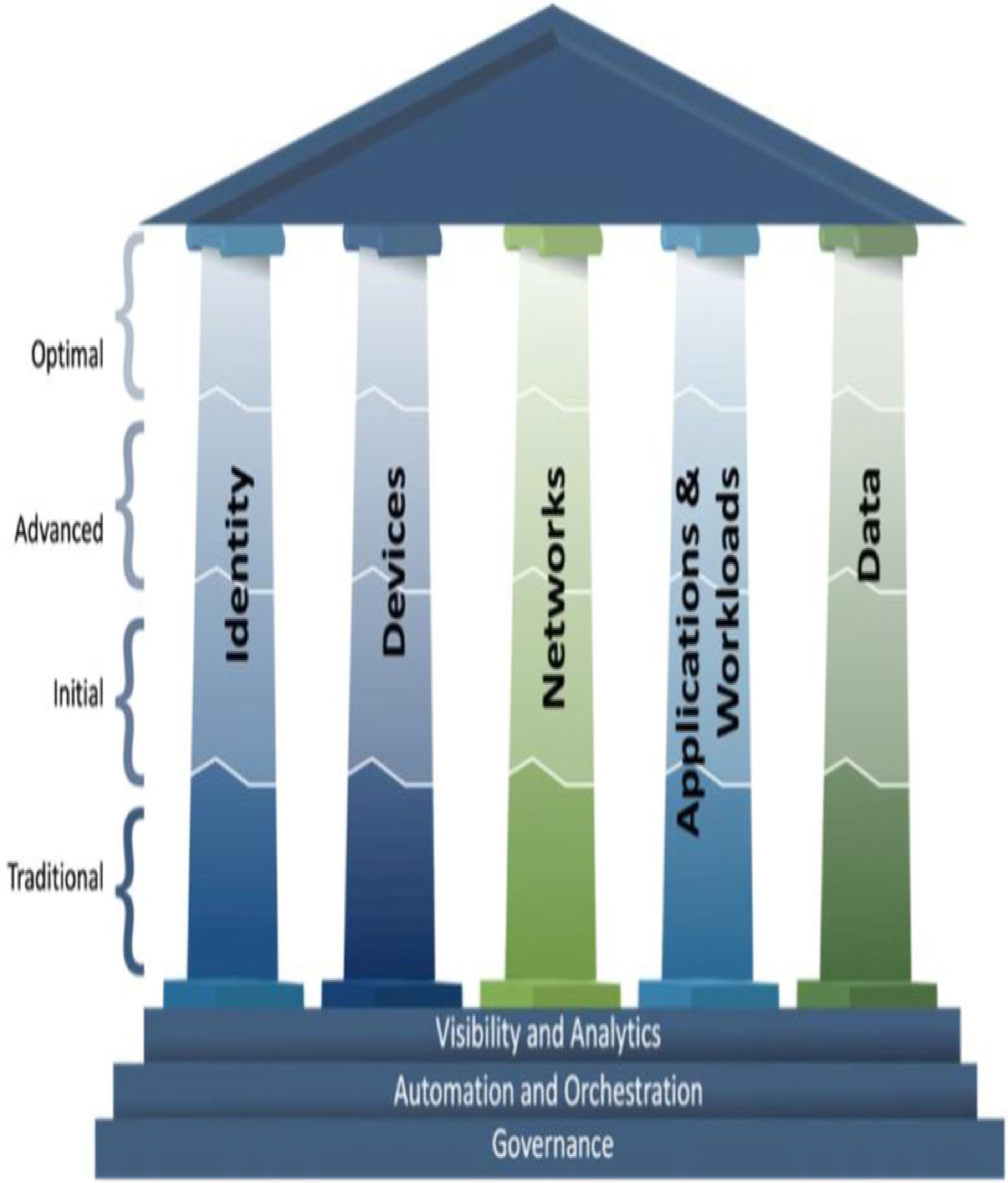


Figure 1-4 *CISA Zero Trust Maturity Model*

6. ISO/IEC 27001 (Information Security Management Systems)

Although ISO/IEC 27001 is not explicitly a zero trust standard, it offers a comprehensive framework for implementing security controls that effectively complement zero trust principles. This standard emphasizes the establishment, implementation, maintenance, and continuous improvement of an organization's information security management system (ISMS). By promoting key principles such as least-privilege, continuous monitoring, and access control, ISO/IEC 27001 aligns closely with the foundational concepts of zero trust, thereby supporting organizations in strengthening their security posture and minimizing risk.

Link: https://www.iso.org/standard/27001

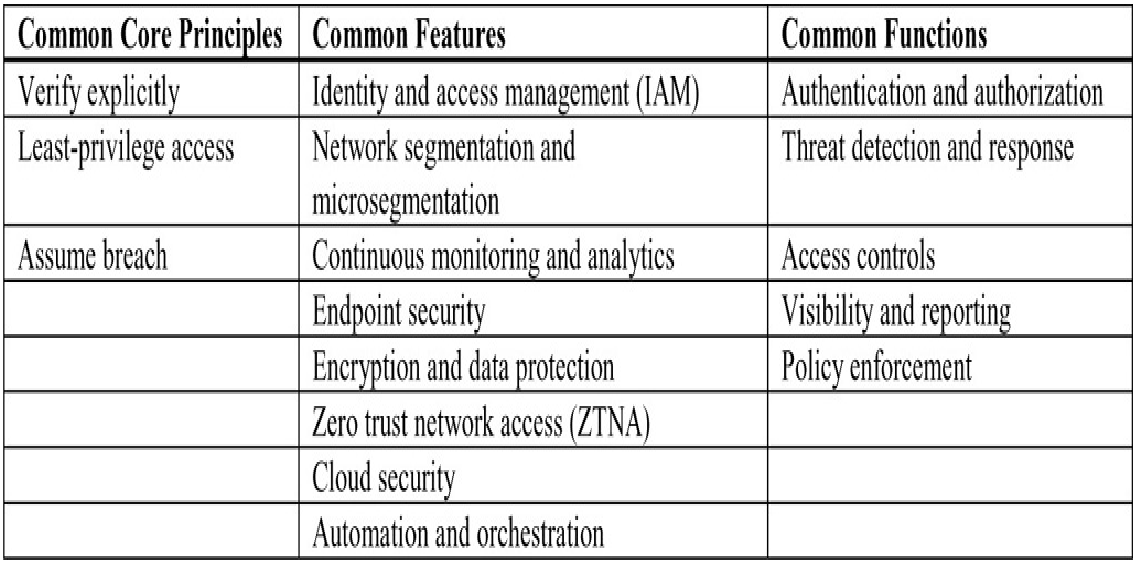
7. **National Cyber Security Centre (NCSC) Zero Trust Architecture Design Principles**

   The NCSC offers a set of design principles aimed at guiding organizations in the implementation of zero trust architectures. These principles focus on critical areas such as identity and access management, ensuring that only verified users and devices gain access; device security, to maintain the integrity of all connecting hardware; network security, which involves safeguarding all network communications; and comprehensive monitoring to detect and respond to potential threats in real time. By adhering to these principles, organizations can effectively build a robust zero trust framework that enhances overall security and resilience.

   Link: https://www.ncsc.gov.uk/collection/zero-trust-architecture

Even though the various standards and frameworks differ somewhat in their approach, all of them have a common set of criteria. The common features and functions of zero trust standards and frameworks, such as those from NIST, Forrester, and others, revolve around a set of core principles and security practices designed to protect modern IT environments. Table 1-1 outlines the key features and functions shared across these various zero trust standards and frameworks:

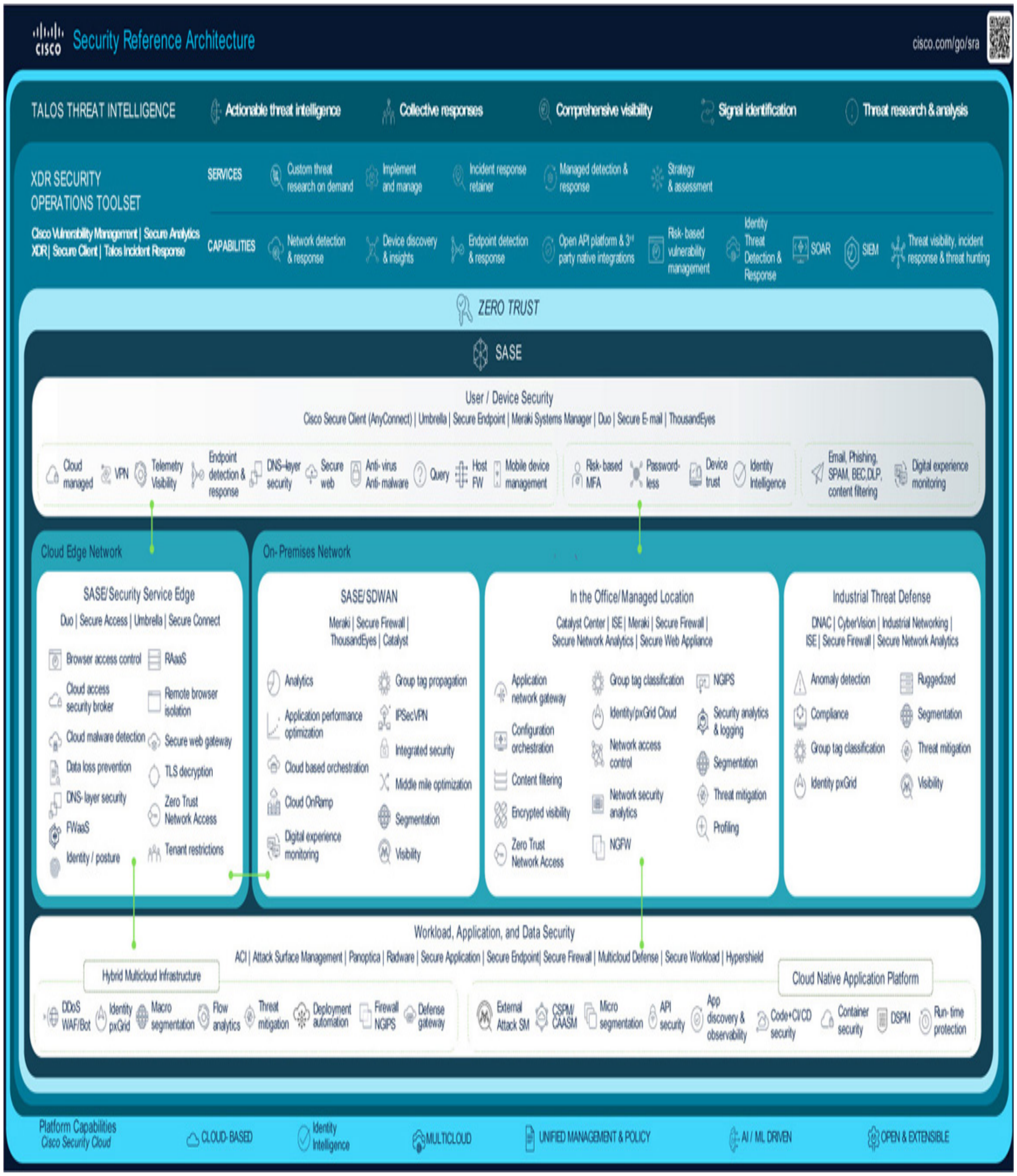
Table 1-1 *Shared Features and Functions Across the Various Zero Trust Standards*



Common Core Principles	Common Features	Common Functions
Verify explicitly	Identity and access management (IAM)	Authentication and authorization
Least-privilege access	Network segmentation and microsegmentation	Threat detection and response
Assume breach	Continuous monitoring and analytics	Access controls
	Endpoint security	Visibility and reporting
	Encryption and data protection	Policy enforcement
	Zero trust network access (ZTNA)	
	Cloud security	
	Automation and orchestration	

In essence, the commonalities across the various zero trust standards emphasize a holistic, continuous, and adaptive approach to security. By focusing on identity verification, strict access controls, data protection, continuous monitoring, and automated responses, all these standards provide a robust framework for implementing zero trust principles. Selecting the appropriate zero trust standard or framework for your organization involves a comprehensive evaluation of various factors to ensure that the chosen approach aligns with your specific security needs, regulatory requirements, and operational goals.

Taking into account the various industry security frameworks such as NIST, CISA, and DISA, Cisco developed its own security reference architecture (SRA), as shown in Figure 1-5. The SRA provides a comprehensive overview of the Cisco Secure portfolio, detailing commonly deployed use cases and recommended capabilities within an integrated framework. Cisco’s approach centers around workforce, workplace, and workload and is structured around these main components: (1) threat intelligence, (2) security operations toolset, (3) user/device security, (4) network security (both cloud edge and on-premises), and (5) workload, application, and data security. This architecture is designed to guide organizations in mapping out their security journey, highlighting key selling motions like zero trust, Secure Access Service Edge (SASE), and extended detection and response (XDR), and emphasizing the advantages of Cisco’s integrated security solutions.



Security Reference Architecture

cisco.com/go/sra

TALOS THREAT INTELLIGENCE

Actionable threat intelligence

Collective responses

Comprehensive visibility

Signal identification

Threat research & analysis

XDR SECURITY
OPERATIONS TOOLSET

Cisco Vulnerability Management | Secure Analytics
XDR | Secure Client | Talos Incident Response

SERVICES

Custom threat research on demand

Implement and manage

Incident response retainer

Managed detection & response

Strategy & assessment

CAPABILITIES

Network detection & response

Device discovery & insights

Endpoint detection & response

Open API platform & 3rd party native integrations

Risk-based vulnerability management

Identity Threat Detection & Response

SOAR

SIEM

Threat visibility, incident response & threat hunting

ZERO TRUST

SASE

User / Device Security

Cisco Secure Client (AnyConnect) | Umbrella | Secure Endpoint | Meraki Systems Manager | Duo | Secure E-mail | ThousandEyes

Cloud managed

VPN

Telemetry Visibility

Endpoint detection & response

DNS–layer security

Secure web

Anti-virus Anti-malware

Query

Host FW

Mobile device management

Risk-based MFA

Password-less

Device trust

Identity Intelligence

Email, Phishing, SPAM, BEC,DLP, content filtering

Digital experience monitoring

Cloud Edge Network

SASE/Security Service Edge

Duo | Secure Access | Umbrella | Secure Connect

Browser access control

RAaaS

Cloud access security broker

Remote browser isolation

Cloud malware detection

Secure web gateway

Data loss prevention

TLS decryption

DNS-layer security

Zero Trust Network Access

FWaaS

Identity / posture

Tenant restrictions

On-Premises Network

SASE/SDWAN

Meraki | Secure Firewall |
ThousandEyes | Catalyst

Analytics

Group tag propagation

Application performance optimization

IPSecVPN

Integrated security

Cloud based orchestration

Middle mile optimization

Cloud OnRamp

Segmentation

Digital experience monitoring

Visibility

In the Office/Managed Location

Catalyst Center | ISE | Meraki | Secure Firewall |
Secure Network Analytics | Secure Web Appliance

Application network gateway

Group tag classification

NGIPS

Identity/pxGrid Cloud

Security analytics & logging

Configuration orchestration

Network access control

Segmentation

Content filtering

Threat mitigation

Network security analytics

Encrypted visibility

Profiling

Zero Trust Network Access

NGFW

Industrial Threat Defense

DNAC | CyberVision | Industrial Networking |
ISE | Secure Firewall | Secure Network Analytics

Anomaly detection

Ruggedized

Compliance

Segmentation

Group tag classification

Threat mitigation

Identity pxGrid

Visibility

Workload, Application, and Data Security

ACI | Attack Surface Management | Panoptica | Radware | Secure Application | Secure Endpoint| Secure Firewall | Multicloud Defense | Secure Workload | Hypershield

Hybrid Multicloud Infrastructure

DDoS WAF/Bot

Identity pxGrid

Macro segmentation

Flow analytics

Threat mitigation

Deployment automation

Firewall NGIPS

Defense gateway

Cloud Native Application Platform

External Attack SM

CSPM/ CAASM

Micro segmentation

API security

App discovery & observability

Code+CI/CD security

Container security

DSPM

Run-time protection

Platform Capabilities
Cisco Security Cloud

CLOUD-BASED

Identity Intelligence

MULTICLOUD

UNIFIED MANAGEMENT & POLICY

AI / ML DRIVEN

OPEN & EXTENSIBLE

Figure 1-5 *Cisco Security Reference Architecture*

People, Processes, and Technology

Implementing zero trust goes beyond technology and requires a holistic approach that also encompasses people and processes. A comprehensive zero trust strategy needs to align people, processes, and technology to ensure that security is not dependent on perimeter defenses alone but focuses on continuous verification and minimization of risks across the network. In summary, a successful zero trust implementation across people, processes, and technology involves several key components:

- **People:**
  - Skills and understanding and training about the human element in security, and why certain measures need to be taken and applied to remain secure.
  - Clear definition of roles and which needs the individual has in terms of access to systems, sites, and locations both digitally and physically.
  - Ability to clearly determine identity, both physically and in their digital and online presence, utilizing modern and strong techniques.
- **Processes:**
  - The speed of impact to a business's reputation is rapid, and containment is not always an option. Therefore, the need to automate the response to a threat, maintain the right level of observability, and monitor and apply the correct level of privileges and controls is essential.
- **Technology:**
  - Modern and up-to-date technology is needed to ensure security in a corporation. Simply applying a security architecture that would have been valid 15 years ago and expecting that it will be secure will most likely not end well. Therefore, it is important to use the latest evolving technology and techniques such as AI-based observability and analytics, strong encryption, micro- and macrosegmentation, multifactor authentication, endpoint detection and response, and other modern techniques.

People

Social engineering continues to be one of the main vectors of attacks and security breaches into businesses today, in contrast to the scams of the early

2000s when your long-lost nephew you never knew about turns out to be an African prince who needs your support with a money transfer. Today’s attacks are a lot more elaborate, and in many circumstances such as phishing, the crafted messages are almost impossible to differentiate without inspection from the real thing. Newer scams are also becoming prevalent, such as the use of deepfake video to pretend to be an individual to breach a system or financial gain. An exploit of this sort was applied by a malicious party that created a deep fake imposter of the CFO and other employees at the British design firm Arup, resulting in $25 million being transferred. Further breaches of this sort are being seen in the areas of financial fraud, with the technical capabilities improving on a frequent basis.

To prepare employees for such challenges, it is key to ensure that a valid and ongoing plan for education is in place—one that includes trainings which are updated, improved, and repeated on a regular basis. Modern efforts to ensure that employees are indeed paying attention to the trainings may include ad hoc (unscheduled) phishing attempts or social engineering exercises, to train employees to understand what to respond to. Part of this training should include a culture of openness when reporting threats; it should not be considered a negative thing to share something that an employee considered unusual that they observed with a contractor or even within their own team or by their management. This sort of activity should be championed and welcomed to ensure a safe and secure organization that protects the organization from financial or reputational harm.

Employees, externals contractors, and partners alike should all be aware of what the zero trust strategy is for the organization and apply the same safeguards, including least-privilege access and multifactor authentication for their respective workforce. Many breaches are the result of inside attacks, or malfeasance on behalf of a party that was trusted. Setting the security apparatus in place, which includes an irrefutable audit trail, with the right levels of identity access management helps in lowering the likelihood of a malicious activity taking place. When the policy is known and understood across the board, an individual considering a malicious act may perhaps think twice. Would individuals steal an item from a shop knowing that they are actively being recorded on camera? Perhaps but the likelihood is certainly reduced.

A small company is often comparable to a small town. The town sheriff may work days at the local gas station, do part-time work as a handyman, and in emergencies volunteer in the town’s fire department. As companies grow larger, the allocation and distribution of roles tend to overlap less, to the point that dedicated roles, teams, departments, or business entities are created and focused on a specific action, task, or job. The setting of clear roles and responsibilities not only fosters a more efficient organization but also helps in ensuring that the right level of security rules can be applied. Should individuals who work at reception have access to the public financial forecasting information? Does the network team need access to the active directory or identity systems? Do financial controllers need to be able to log in to routers and switches? A clear understanding of roles, responsibilities, and tasks allows for these definitions to be applied on an identity or identity group basis to ensure that zero trust principles can properly be exercised via role-based access control.

Processes

Having the right organizational processes in place is essential, and just as important is ensuring that those processes are clear, concise, and understandable to ensure that everybody is on the same page. Ambiguity can lead to misinterpretations and, eventually, missteps in upholding the correct security practices within an organization. Therefore, ensuring that processes are well-documented and defined can help ensure that the subsequent information systems that depend on the defined processes are calibrated, configured, and deployed in the right manner to match up with the vision of the company.

Common processes within an organization can be around the implementation of strong identity and access management practices, such as the length and complexity of a username and passwords, the maximum period between updates, and the mandatory use of certificates for infrastructure components and how often they should be updated or changed.

Similar processes may exist for how to ensure that IoT systems that are inherently insecure are procured only by vendors that ensure a minimum level of encryption or authentication capabilities. Also, there may be processes for how such systems are kept isolated and separate from other critical systems in scenarios that they do not meet corporate requirements.

For supporting systems, where human operators are in the loop, processes to mandate multifactor authentication and single sign-on are very common. Those processes are coupled with mandatory regular audits, updates, and checks to ensure that the software used by the operators complies with needed access policies and authorization rulesets to reduce the likelihood of a breach.

Ensuring the monitoring and baselining of expected behavior is important when attempting to detect an anomaly that may be occurring in an architecture. Process definitions are needed for what should be collected, the retention period for that data to allow for longer post-mortem data traversal, and under certain circumstances, to satisfy governmental mandates. Having such definitions clearly defined for information systems and infrastructure devices within an IT domain ensures that the right controls are available from the beginning, and integration into security information and event management (SIEM) systems for analysis can easily take place.

Processes within a network should extend to include minimal requirements for access to resources that describe which regulations (such as GDPR and HIPAA) need to be met for compliance within a particular region. Under certain circumstances, totally independent and separate architectures may be required to ensure that data remains within a country to meet these specific regional needs.

Technology

Maintaining and staying up-to-date with the latest technology to ensure that a secure estate is deployed sometimes may feel like a game of whack-a-mole, with constant advancements in technology and threats making their way into the security space. It is easy to feel overwhelmed.

Security concepts such as endpoint security are often leveraged in conjunction with network access control, which can result in endpoint isolation or restriction into limited connectivity domains. This isolation is often achieved through macro- and microsegmentation and dynamic policy enforcement. Client-side posture agents and mobile device management (MDM) capabilities can also help augment the edge of the network, supporting dynamic privilege allocation, which in turn can help reduce lateral movement-based attacks.

Beyond the network and the clients themselves, further tools to secure communications between network junction points, such as quantum-resistant encryption, data loss prevention, and classification tools, can help raise the robustness and security of critical data within a corporate architecture. For private and public clouds, security tools such as cloud native application protection platforms can ensure that cloud workloads are secured and potential threats are identified and can be mitigated. Such tools become extremely helpful when attempting to deal with disparate hyperscaler and on-premises estates.

In addition to tools that help ensure a better cloud native application security posture, the automation of security tasks through security orchestration, automation, and response (SOAR) tools can lower the operational overhead of an organization when attempting to respond to incidents and threats, based on analytics that can be linked to trend-based benchmarking, or anomaly detection based on graph-based prediction of user and usage patterns. This sort of tool is becoming much more heavily augmented in recent years through predictive analytics capabilities, which are awarded through artificial narrow intelligence (ANI) assessment and analysis.

On-Premises vs. Cloud

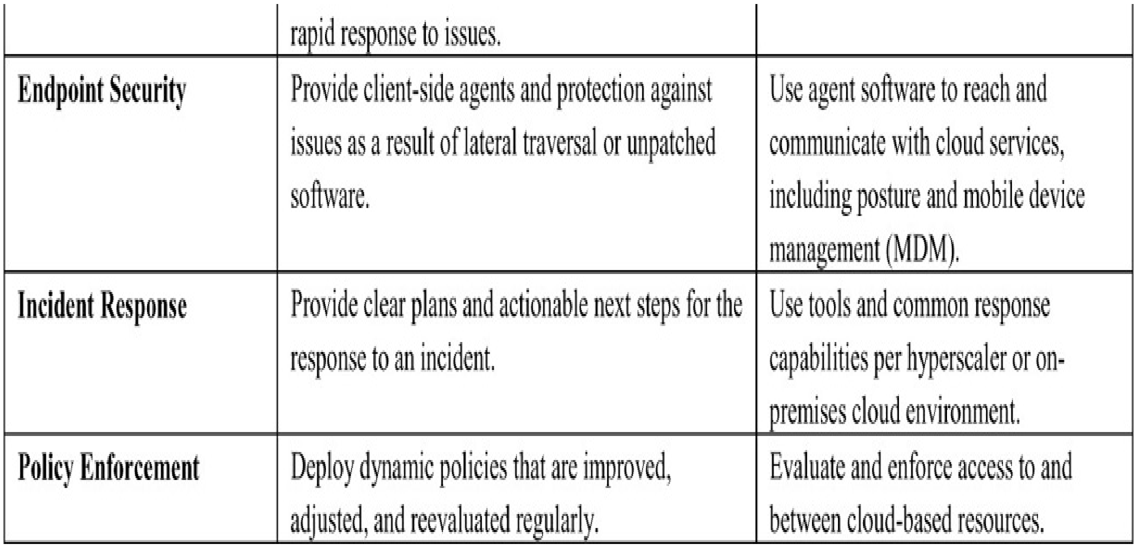
When we look into the physical security of premises, it would be very uncommon for individuals to attempt to secure their house with monitor lighting only in the backyard while disregarding the front yard and garage, which also represent entry points into the building. Taking such shortcuts with technology is clearly not advocated. When we're looking at zero trust principles, they need to be applied consistently within an estate, regardless of whether it is on-premises, in the data center, while using a cloud-based service, or when accessing and interfacing with core business data on a Software-as-a-Service (SaaS)–based offering.

Ensuring that an equal level of security and restriction is applied both on-premises and in the cloud and in hybrid scenarios is important; however, the way that this approach is pursued may not be replicable across the board. As with each technology platform, the software version of a vendor's solution, and their unique combination of different features are often applied to achieve a Zero Trust deployment. This applied combination of functionalities and vendor solutions should represent an end-to-end means to both secure communications and assets. The application of the zero trust principles, ensures that least-privilege access is applied, and persistent monitoring and verification are always in place to achieve the needed visibility and outcomes.

In the following sections and Table 1-2, we further explore how on-premises versus cloud principles are applied to the concepts of zero trust.

Table 1-2 *Zero Trust Recommendations On-Premises vs. Cloud*

Aspect	On-Premises Recommendations	Cloud Recommendations
Explicit Verification	Identify systems such as active directory, LDAP, multifactor authentication, and group- and role-based access control.	Use hyperscaler-specific identity management services and federated services to provide a common identity-based approach to common security and role-based access capabilities.
Least-Privilege Access	Limit access to network resources via macro- and microsegmentation, providing only access required for a minimum needed function, based on system and role. This can be combined with policy-based rule decisions through tools such as TACACS+ or RADIUS.	Use just-in-time access and resource-level controls and cloud identity and access managements systems.
Segmentation	Use dynamic policy and network access control to steer and deploy segmentation, per user, role, service, and endpoint.	Deploy isolated cloud resources together with policy-based tools like Cilium to provide service-based separation.
Monitoring	Use centralized telemetry from client, server, and infrastructure systems toward SIEM systems, together with standardized intrusion detection and prevention architectures	Use viable application log levels and apply security configurations within each cloud architecture that is used, in conjunction with a means to centrally validate and audit.
Encryption	Apply data in transit, data at rest, and data in memory to the correct frameworks and systems that require selection to meet organizational needs. This can include securing secrets through the use of tools like HashiCorp Vault or OpenBao. Using VPN technology with post-quantum keying and updated algorithms and methods can also improve security.	Use the shared responsibility model, plus the correct levels of data classification and encryption to meet regulatory needs, and protect key business-relevant data. Augmenting cloud offerings with post-quantum Transport Layer Security (TLS) can further harden deployments.
Automation	Apply techniques to ensure that the appropriate automated actions occur through the use of SOAR and other system-level automation to allow for a	Embed security policies and logic within cloud native automation as a key aspect to ensure a secure architecture.



	rapid response to issues.	
Endpoint Security	Provide client-side agents and protection against issues as a result of lateral traversal or unpatched software.	Use agent software to reach and communicate with cloud services, including posture and mobile device management (MDM).
Incident Response	Provide clear plans and actionable next steps for the response to an incident.	Use tools and common response capabilities per hyperscaler or on-premises cloud environment.
Policy Enforcement	Deploy dynamic policies that are improved, adjusted, and reevaluated regularly.	Evaluate and enforce access to and between cloud-based resources.

Explicit Verification

In on-premises environments, identity solutions like Microsoft Active Directory and LDAP are crucial for mapping users to roles and business functions, thus ensuring proper authorization to network segments, resources, and potentially bandwidth allocations. These systems facilitate role-based access control and can incorporate multifactor authentication for enhanced security, especially when accessing critical resources or privileged accounts. In cloud environments, the principles of identity verification remain consistent with zero trust frameworks, emphasizing explicit identity verification to secure networks and critical systems. Common cloud identity solutions include Google Identity Platform, Amazon Web Services (AWS) IAM, Alibaba Cloud IDaaS, and Microsoft Azure Entra ID, while vendor-independent services like Okta and Oort also play a role in user verification. Despite differing infrastructures, both on-premises and cloud systems prioritize rigorous identity management to maintain security integrity.

Least-Privilege Access

In on-premises environments, the principle of least-privilege access is applied by granting users access only to the resources and network segments necessary for their job roles, sometimes restricted to specific time periods. This involves using authorization rules through protocols like RADIUS for network access or TACACS+ for device interactions, and extends to applications and confidential document repositories. In contrast, cloud environments also emphasize least-privilege access, but the expansive capabilities and flexibility of cloud services introduce unique risks, such as insider threats leading to unauthorized activities like bitcoin mining, which can cause significant financial damage. Properly controlled access to cloud resources, such as virtual machines, databases, and machine learning tools, is crucial. Cloud identity and access management (IAM) policies should be finely tuned to limit access to resources like S3 Buckets, Azure and GCP Cloud Storage, compute services like EC2 or Cloud Engine, and cloud graphics processing units (GPUs). Additionally, cloud environments often employ just-in-time and temporary access solutions to prevent unnecessary persistent access, aligning with zero trust principles.

Segmentation

In on-premises environments, effective segmentation involves ensuring that endpoints have only the necessary access to target systems, such as allowing OT systems to communicate solely with a central controller or restricting corporate laptops to nondeveloper network environments. This concept has evolved from static IP subnetting to more dynamic and automated solutions like network access control and software-defined architectures, such as EVPN and SD-Access, making network segmentation more flexible and widespread. In cloud environments, segmentation is equally critical to prevent breaches, especially with containers running different microservices on shared nodes. Cloud segmentation uses virtual private clouds or virtual networks to separate resources and implement differential access policies, employing security groups and network access control lists. Tools like Cilium (an open-source project that enhances networking and security for containerized applications using eBPF) further enhance segmentation and network control by deploying policies based on service identities, abstracting user-accessible systems from the underlying infrastructure, and protecting critical resources.

Continuous Monitoring and Threat Detection

In on-premises environments, security relies on continuously sending relevant data from client and server systems, network devices, and security systems to a security information and event management (SIEM) system. This setup facilitates the detection and analysis of potential threats by aggregating and assessing security information. Intrusion detection and prevention systems, operating transparently, help identify unusual traffic patterns by comparing them against benchmarks of normal network behavior. By monitoring network flow data, deviations from expected user behavior, such as a sudden increase in data transfers, can raise security alerts. In cloud environments, monitoring application behavior and security rules is crucial, with major providers like Google, AWS, and Microsoft offering integrated monitoring tools like Cloud Watch and Azure Monitor. The cloud operates on a shared responsibility model, dividing security duties between the provider and the customer. This model prompts some organizations to use third-party security solutions, such as Splunk or Wiz, to enhance monitoring and threat detection, ensuring impartiality and comprehensive security oversight.

Encryption and Data Protection

In on-premises environments, data encryption has long been a foundational security practice, evolving with stronger algorithms and hardware or software capabilities to encrypt data in transit, at rest, and sometimes even in memory. Advanced encryption solutions, such as hardware security modules and secret management tools like HashiCorp Vault, are used to protect sensitive data and keys, while emerging technologies like blockchain-based data loss prevention offer additional layers of security, albeit not widely adopted. In cloud environments, providers typically offer built-in encryption capabilities, such as AWS’s hybrid post-quantum TLS in its key management service, with each hyperscaler providing its own encryption and key management solutions. Cloud customers can further enhance data protection by deploying specific encryption tools tailored to their workloads, either through cloud marketplaces or third-party software requiring separate installation, allowing for a flexible and customizable approach to security in the cloud.

Automated Response and Orchestration

In on-premises environments, the swift response to potential threats is crucial for minimizing the impact of security incidents. This is achieved through automation coupled with robust rulesets, enabling enforcement actions across security architectures. The deployment of security orchestration, automation, and response (SOAR) systems, along with Infrastructure as Code (IaC) pipelines for software upgrades and patches, is becoming standard practice for handling potential breaches effectively. In cloud environments, automation principles have been integral from the start, with hyperscalers prioritizing API-first development followed by user experience enhancements. This approach facilitates a programmability-first strategy for updates, patching, and configuration, allowing for agile responses to security incidents. With proper security hygiene and rulesets, cloud environments can execute automated A/B testing and phased deployment for security updates, reducing risk during provisioning changes.

Endpoint Security

In on-premises environments, securing endpoints involves collecting critical data, preventing lateral movement, and responding to malicious activities that might arise from compromised endpoints. This is often achieved through network access control and mobile device management systems, along with modern endpoint detection and response (EDR) architectures that deploy agents for centralized monitoring and ensure timely patch and update deployment, sometimes limiting access or speed until patches are applied. In cloud environments, endpoint security focuses on client-side agents connecting to security brokers or gateways, providing secure access to hyperscaler resources or SaaS services. This setup typically limits peer-to-peer communications, favoring a centralized traffic model. Posture and mobile device management capabilities, common in on-premises security, are also prevalent in cloud communications, ensuring consistent security standards across both environments.

Incident Response and Recovery

In on-premises environments, effective incident response requires thorough preparation to minimize impact, reduce response time, and clarify ownership and execution of tasks not yet automated. This includes capturing the post-incident state for investigation and implementing disaster recovery procedures to restore systems to their last known good state. Regular testing of backup restoration supports a robust business continuity plan. In cloud environments, incident response similarly depends on well-defined action plans and disaster recovery strategies, particularly in multicloud settings with distributed workloads. Each cloud provider offers security tools to aid in responding to incidents, but organizations must understand deployment nuances and ensure comprehensive consideration of all affected systems and breach points to prevent recurrence.

Policy Enforcement

In on-premises environments, policy enforcement must dynamically adapt to evolving security standards based on user roles, endpoint types, patches, and usage patterns to ensure systems remain secure and do not become vulnerable entry points. Continuous evaluation of the technical estate against changing requirements is essential to maintain security integrity. In cloud environments, similar security principles and policy enforcement are critical, with a focus on zero trust strategies that limit access to necessary resources. Cloud deployments often span multiple regions, necessitating compliance with regulations like GDPR and HIPAA, which require specific considerations to maintain a secure and compliant deployment.

Hybrid Environment Recommendations

In today’s digital landscape, many startups launch their businesses entirely using cloud resources, often without a physical office or garage. Conversely, established brick-and-mortar companies frequently maintain significant on-premises environments. To balance flexibility and cost-effectiveness while ensuring business continuity, many organizations adopt a hybrid deployment model. This model involves a mix of on-premises and cloud infrastructures, sometimes utilizing multicloud strategies. Such distributed approaches can offer benefits like vendor negotiation leverage and reduced risk of price gouging, but they also present security challenges. A major risk is that siloed teams may operate independently without coordinated communication, a legacy approach that can lead to security vulnerabilities.

Effective communication and integrated security architectures are essential to mitigate these risks. Organizations must consider end-to-end security solutions, ensuring consistent identity management, encryption, and dynamic policy enforcement across all environments. This often involves federated identity and access management to reduce complexity and maintain security across hybrid systems. Collecting telemetry data and logs in a centralized SIEM system provides a comprehensive overview, facilitating the correlation of incidents between cloud and on-premises systems. Consistent security and monitoring help avoid gaps that could lead to confusion or missed security incidents.

In hybrid and multicloud deployments, platform engineering approaches are increasingly used to standardize security rules and telemetry activation, removing this burden from application teams. Endpoint security must also be consistent, focusing on identity and the principle of least access to minimize the impact of potential compromises. As hybrid architectures remain prevalent and data repatriation gains traction, implementing robust security policies and incident response plans is crucial for maintaining secure and scalable operations.

Hybrid architectures are not going away. In recent years, repatriation of data and workloads is becoming more prevalent, and as a result, ensuring that the right policies and plan of action take in the context of security incidents is fundamental in maintaining a secure and scalable deployment.

Security Certifications

The number of global regulations and standards that exist allow for a good level of public discourse but at the same time can lead to confusion around which rules to follow. In the domains of compliance, security, and the mitigation of risk for your organization, there is a need to follow and stay up-to-date with key standards and certifications that exist in the field.

The following list is not exhaustive and has been curated to provide an overview of some of the most common certifications and standards that are being followed in the field, in defense, service provider, and enterprise industry verticals:

- **Federal Information Processing Standards (FIPS):** FIPS was a consequence of the Federal Information Security Management Act in the United States, which was enacted in December 2002. The subsequent standards from NIST were developed primarily with a focus for nonmilitary use, with a focus on use by US federal agencies; however, these standards have been heavily adopted by organizations globally as a means to orient information systems toward a secure architecture.

  Subsequently, many technology vendors provide FIPS-compatible modes or FIPS-compliant hardware for their products to allow for their use with aligning government agencies. The deployment of such systems tends to automatically disable weak encryption algorithms and other insecure capabilities that would tend to exist in a product or portfolio.

  While there are many Federal Information Processing Standards, the most noteworthy one to mention is FIPS-140-2: Standard describing both security and encryption required for IT systems. The FIPS focus on encryption of data falls heavily around the need to secure the communication of data required in zero trust architectures.

- **Cloud Security Alliance:** The Cloud Security Alliance provides a certification program focused on SaaS and cloud-based offerings. A significant number of global corporations collaborate with CSA on ensuring that their offerings meet the standards set out for deployment hygiene and security best practices.

  While the principles outlined by the CSA are not all focused on zero trust, their activities provide a key foundation toward ensuring that the estate follows the right practices toward achieving an increased level of security in the domain of continuous monitoring and security best practices.

- **ISO / IEC 27001: Information Security Management:** The International Organization for Standardization created the 27001 standard focused on information security management systems. The purpose is to create vertical agnostic guidance on data security and risk management in the domain of cyber security. This standard delves deeply into a diverse range of factors associated with organizations, ranging from internal and external influences to how organizations appropriately manage risk. In the context of a zero trust deployment, an understanding of the technical considerations and the people and process challenges that you could be faced with, both internally and externally, is critical.
- **National Institute of Standards and Technology:** NIST is an agency that focuses on standards for technology. The agency itself belongs to the US Department of Commerce. As part of the agency’s remit, it provides special publications, which provide guidance and information that are key to the domain of information security. There are several publications with specific relevance to security and zero trust; they include NIST SP 800-207, SP 800-53, and SP 800-63.
- **System Organization Controls 2 (SOC 2):** This popular audit type can be applied to service organizations to ensure their veracity in properly protecting confidential data that belongs to their customers. SOC 2 is based around five key categories: security, availability, confidentiality, processing integrity, and privacy.
- **Cybersecurity Maturity Model Certification (CMMC):** The CMMC framework was developed under the US Department of Defense to better support the ability to determine the level of maturity that a given department, unit, or organization has achieved in its respective deployment of security standards defined by NIST. The certification of an organization following the CMMC is typically performed by an external auditing party, so as to avoid a conflict of interest.

While zero trust and the CMMC certification are not completely analogous to one another, the segmentation and principles of least access that are exercised in zero trust architectures go a long way toward achieving a positive certification result.

- **Payment Card Industry Data Security Standard (PCI DSS):** In retail environments, the requirement to ensure that credit and debit card information from customers is managed securely is critical. The PCI DSS was created to ensure that the millions of shops, stores, hotels, and other retail spaces making use of payment card technology have their data managed securely.

  This standard outlines the needs for access control methods associated with the technology, including the monitoring, securing, and auditing of sensitive data. Currently, this standard is applied to a broad range of credit card suppliers, including Visa, Mastercard, and American Express.

  Taking the principles of zero trust seriously within this domain, at all stages of the transaction process, ensures that the payment card industry remains secure, and customers can confidently perform transactions without losing trust with electronic payment technologies.

- **Health Insurance Portability and Accountability Act (HIPAA):** Private health data is one of the most sensitive and personal types of electronic data that exists for most individuals. Depending on the country where this data is being handled, its landing in the wrong hands can lead to a number of negative repercussions for the individual, ranging from discrimination to increased health insurance premiums to challenges with employment.

  To ensure that such data remains secure, the Health Insurance Portability and Accountability Act was introduced. It provides key focus in the domain of zero trust, around the protection and access control of sensitive health and patient data, with ongoing compliance auditing and monitoring of the data, to control access breaches and tampering of the respective data.

Summary

Zero trust is a cybersecurity strategy that fundamentally shifts the traditional approach to network security by assuming that threats can exist both inside and outside the network perimeter. Originating from principles articulated by John Kindervag at Forrester Research in 2010, zero trust advocates for "never trust, always verify," meaning that no entity, whether inside or outside the network, is trusted by default. Various standards, including NIST's Special Publication 800-207, provide comprehensive and structured frameworks, guiding organizations in implementing zero trust principles. These frameworks emphasize robust identity verification, least-privilege access, continuous monitoring, and granular access controls to ensure that every request for network access is authenticated, authorized, and encrypted. These standards have since evolved into best practices, with certifications like CISA's Zero Trust Maturity Model and ISO/IEC 27001 offering benchmarks to measure and certify zero trust compliance, helping organizations structure robust and actionable security models.

Implementing a zero trust strategy offers numerous benefits to organizations. It significantly reduces the risk of data breaches by limiting lateral movement within the network, thereby containing potential threats. Additionally, by continuously monitoring and validating user identities and device health, organizations can swiftly detect and respond to anomalies. Zero trust is also future proof, offering resilience against evolving cyber threats, including those posed by quantum computing. As quantum computing advances, it could potentially break traditional encryption methods, but zero trust architecture's emphasis on microsegmentation and continuous validation of users and access rights can help mitigate these risks. Therefore, adopting zero trust not only enhances current security postures but also prepares organizations to face future cybersecurity challenges. By adopting a zero trust approach, organizations can position themselves to safeguard critical assets both now and in the future, regardless of how attacks evolve.

References

1. John Kindervag, *No More Chewy Centers: The Zero Trust Model Of Information Security*: https://www.forrester.com/report/No-More-Chewy-Centers-The-Zero-Trust-Model-Of-Information-Security/RES56682

2. General Data Protection Regulation (GDPR): https://eur-lex.europa.eu/eli/reg/2016/679/oj

3. Health Insurance Portability and Accountability Act (HIPAA): https://www.hhs.gov/hipaa/index.xhtml

4. Google BeyondCorp: https://cloud.google.com/blog/topics/developers-practitioners/zero-trust-and-beyondcorp-google-cloud

5. NIST 800-207: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

6. CISA Zero Trust Maturity Model: https://www.cisa.gov/sites/default/files/2023-04/zero_trust_maturity_model_v2_508.pdf

7. Institute for Security and Open Methodologies (ISECOM): https://www.isecom.org/OSSTMM.3.pdf

8. Cisco Security Reference Architecture: https://www.cisco.com/c/en/us/products/security/cisco-security-reference-architecture.xhtml#~overview

Chapter 2. Secure Automation and Orchestration Overview

In this chapter, you will learn about the following:

- The evolution of network automation
- Zero trust in automation and orchestration
- API security approaches
- Common automation and orchestration practices
- Self-healing networks with AI/GenAI approaches

Introduction to Automation and Orchestration

In the context of computer networks, automation refers to the use of software and tools to manage, configure, monitor, and optimize network operations with minimal human intervention. It enables faster deployments, improves security, reduces errors, and enhances scalability. We deploy automation for repetitive and complex tasks. If we assign the same work to a person repetitively, the person might lose interest after a few iterations and tend to do mistakes. It is common to automate the regression and sanity test cases in engineering development as the common test cases that need to be executed to validate the basic functionality of the software in the shortest amount of time. In a more complex form, industrial automation uses complex robotic processes on the same line for more precision work, such as welding the frame for a car. In terms of IT networks, automation could mean the following:

- **Managing Configuration:** Automating the onboarding and configuration of network devices such as routers, switches, cloud infrastructure, or any other IT devices

- **Provisioning Network Devices:** Ensuring the right configuration or changes are applied to the network devices based on the intent of the business
- **Monitoring and Reporting:** Continuously monitoring the key performance indicators (KPIs) of the network; generating the alerts and producing reports
- **Enforcing Security:** Implementing automated security policies and controls to contain the threats

Orchestration, on the other hand, is the coordinated management of multiple automated and nonautomated tasks to achieve a desired outcome. Orchestration comes into play by putting multiple automation tasks in a row —referred to as a *workflow*. The orchestration layer sits on the top of automation engines. Centralized controllers like Cisco Catalyst Center, Cisco Catalyst SD-WAN Controller, or Cisco Application Policy Infrastructure Controller (APIC) can be used for domain-specific automation. The benefit of using these central controllers as a middle layer for orchestration is that you don't have to directly reach out to every device and just need to interface with these central controllers. You can apply orchestration to different use cases such as these:

- **Service Orchestration:** This operation involves coordinating the deployment, management, and scaling of the network services, such as virtual networks, firewalls, and load balancers.
- **Workflow Orchestration:** This operation involves the systematic execution of multiple tasks based on the use case, such as bringing up an entire network for a new branch site. It will be a combination of activities like provisioning devices, configuring the devices, identifying dependencies, integrating with the monitoring system, and validating basic functions.
- **Network Function Virtualization (NFV) Orchestration:** Virtualized network functions are gaining popularity, especially in service provider networks. For example, 5G core functions are mostly virtualized. The orchestration engine is responsible for

initiating, configuring, and service-chaining the various NFVs to activate specific services.

Automation and orchestration work together in a unified approach to create a cohesive and efficient network management strategy. Automation handles the individual tasks, and orchestration ensures the tasks are completed in the correct order and validates whether the end service is up and running.

Understanding the roles and benefits of automation and orchestration in IT networking is crucial for building and managing modern, efficient, and secure network infrastructures. These technologies form the foundation for advanced networking paradigms, such as zero trust architectures (ZTAs), where security and efficiency are paramount. Figure 2-1 shows the relation between automation and orchestration.

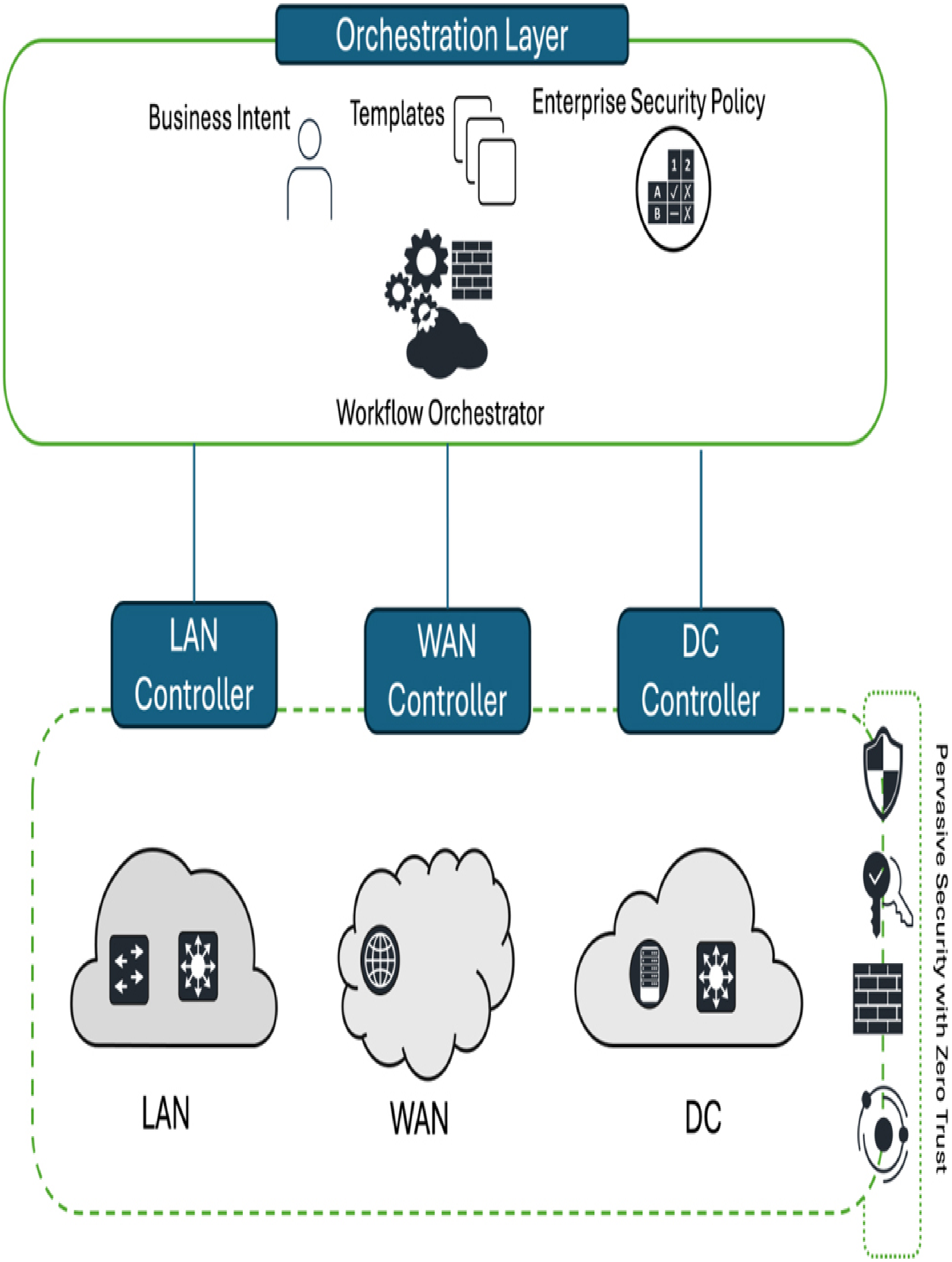


Figure 2-1 *Network Orchestration and Automation Layers*

Zero trust architecture (ZTA) is a security framework that assumes threats can come from both inside and outside a network. Automation and orchestration can play a critical role in enabling ZTA by enforcing consistent security policies across an environment. Also, ZTA should be part of the automation and orchestration tools deployed by any organization because they play a critical role and act as a bridge between the business intent and the actual device. If these tools are compromised, attackers can gain control over large parts of the infrastructure. Access to these systems must be secure.

Let’s revisit some of Cisco’s core principles of zero trust:

- **Verify Explicitly:** Authenticate and authorize users and devices based on all available data points like identity, location, and device health/posture. Multifactor authentication (MFA) is typically used to verify the user’s identity before granting that user access to any part of the network. The concept of continuous MFA (cMFA) that triggers MFA flows at regular intervals or based on trust score thresholds could also be deployed. An example of cMFA is a system that tracks a user’s identity continuously via camera, biometrics, and so on, and continuously validates the activities of the user and maps with the trust score. This score wears down based on time, action, or event by the user. This triggers another MFA for the user. If the user does not respond or if MFA fails, access is restricted or completely blocked. This outcome solves the common problem where one user logs on to a system and passes it on to another user to operate.

- **Use Least Privilege Access:** Provide users access with just-in-time and just-enough access, user-adaptive risk-based access policies, and data protection to reduce the risks. Traditionally, such access was provided based on time-based ACLs. This process was very complicated and had lots of administrative overhead. Such features are now built into the central controllers and cloud-based platforms, allowing secure access to a device based on certain conditions. One such example is Cisco Secure Equipment Access, which allows a remote machine vendor to access a specific device in the industrial environment for a specific duration, using a specified protocol with the ability to continuously monitor, record, and terminate the session anytime. This topic is discussed in more detail in Chapter 21, “Zero Trust in Industrial Manufacturing Vertical.”

- **Assume Breach:** This principle suggests that you should always be prepared in case the network is breached. Design systems to minimize the impact of any threats, typically done using network segmentation. Segmentation needs to be performed at the macro and micro level. Macro levels segment the main systems of the network like corporate users, guest users, third-party vendors, corporate applications, and public applications. Microsegmentation further groups the users and devices based on common access policies and rules required, such as employees from the financial team or development team. Continuous monitoring between different segments is also required to detect and respond to any suspicious activities.

For example, in a microsegmented network, a specialized platform like Cisco Secure Network Analytics monitors the traffic between different segments, isolates the impacted devices, and triggers an incident response workflow. Using Cisco XDR, you can then set up automation workflows that bring together different security solution components such as Cisco ISE, ngFW/IPS, and Cisco Cyber Vision.

Evolution of Network Automation

The evolution of network automation has been driven by the need to manage complex and large-scale networks efficiently. In the early days, computer networks used to be very small with very limited devices. But today networks are gigantic with millions of devices, and more are getting added each day. Service provider networks are adding more nodes and services. Enterprise and industrial customers are using smart devices and IoT sensor networks. These devices use varied types of connectivity, such as Ethernet, Wi-Fi, BLE, LoRA, and 5G. Each technology adds another layer of complexity with its unique protocols, security, and provisioning methods.

Let’s take a closer look at the history of network automation:

- **Stage 1:** During the early days (1980s), devices were manually configured either using a graphical user interface (GUI) or command-line interface (CLI). This approach was good for a small number of devices but posed multiple challenges for wider networks such as the following:

  - There was high potential for configuration error due to manual entry.
  - It was difficult to manage a large network because each device needs some custom configuration such as an IP address.
  - It was a time-consuming task although the use of Notepad or similar editors was common to create the configurations in advance.

- **Stage 2:** During the early 1990s, network administrators started using script-based automation, Perl, and Shell to automate repetitive tasks. Some of the challenges with this approach were
  - Scripts needed to be written for a specific task targeted for a set of devices.
  - Scripts needed to be updated regularly as network configurations changed or new features were added to the devices.
  - It was difficult to integrate the different scripts for cohesive network management.
- **Stage 3:** During the 2000s, network management platforms like Cisco Prime were common. It was a major evolutionary step to advanced network automation. Network admins could now configure and pull the configuration using these SNMP-based NMS systems. But this capability had its own set of challenges, including
  - High initial cost
  - Limited interoperability between vendors
- **Stage 4:** The 2010s started a new era of software-defined networking (SDN), and with it, we had the modern era of automation. These systems came up with native programmability interfaces and simplified network management. These systems allowed integration with other automation and orchestration engines, enabling the development of new network services. This has evolved over some

time to base network automation and orchestration. Products like Cisco Catalyst Center and Cisco Catalyst SD-WAN controller fall into this category. While customers are still adopting this capability, some of the challenges with these systems are as follows:

  - Implementation, operation, and troubleshooting can be complex.
  - There are high upfront costs.
  - There is a lack of trained staff because SDN is still a new concept for a few customers.
  - The learning curve is also very steep. Not many customers have the billable hours to do it while also maintaining an already complex “legacy” enterprise environment.

Another emerging trend is Infrastructure as Code (IaC), which involves managing cloud infrastructure and services through tools and techniques akin to those employed by software developers. This encompasses practices such as version control, peer reviews, automated testing, and deployment pipelines. The IaC methodology is frequently integrated with tools and DevOps practices traditionally used by application developers. This synergy is often referred to as NetDevOps. Detailed discussions of these subjects are provided in later chapters of the book.

- **Stage 5:** The industry is not stopping with SDN and intent-based network automation. At the time of writing this chapter, there have been significant advancements around AI/ML/GenAI concepts. Every automation and orchestration platform is looking at integrating these technologies, thus paving the way for next-generation self-healing networks. Some of the challenges of adopting artificial intelligence and machine learning for predictive analysis and autonomous network management are as follows:
  - A large amount of high-quality data is required for effective AI/ML models.
  - Integrating AI/ML to legacy systems can be complex.

- The network consists of diverse vendor devices, protocols, and configurations, making it challenging to develop AI/ML models.

Network Maturity for Automation

In the previous section, you learned about different stages of automation and orchestration. For practical application, you need to classify an organization into what stage of maturity it is and create a progressive plan to move to the next level of automation and orchestration. Please note that network maturity encompasses a broader range of factors. In this section, the focus is primarily on automation. Because security is essential and must be a fundamental aspect of any maturity framework model, both security and automation are included for the proposed maturity states, as shown in Figure 2-2.

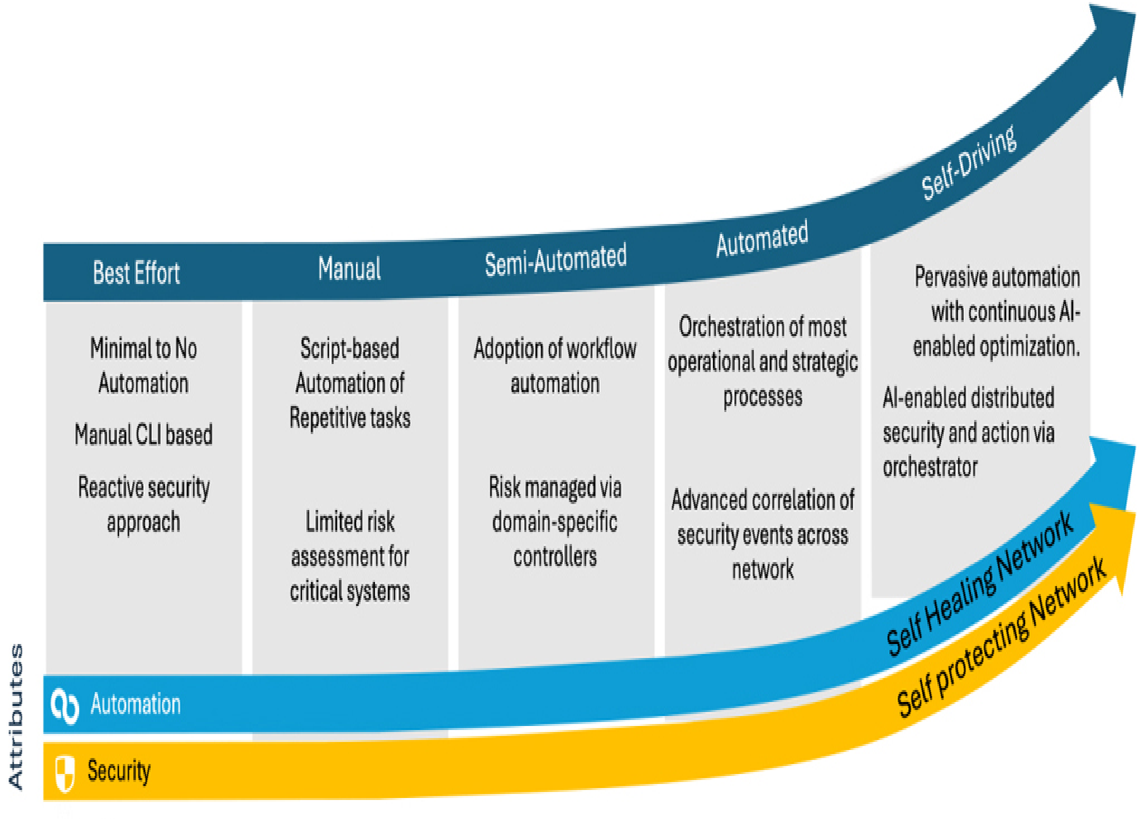

Best Effort
Manual
Semi-Automated
Automated
Self-Driving
Minimal to No Automation
Manual CLI based
Reactive security approach
Script-based Automation of Repetitive tasks
Limited risk assessment for critical systems
Adoption of workflow automation
Risk managed via domain-specific controllers
Orchestration of most operational and strategic processes
Advanced correlation of security events across network
Pervasive automation with continuous AI-enabled optimization.
AI-enabled distributed security and action via orchestrator
Self Healing Network
Self protecting Network
Attributes
Automation
Security

Figure 2-2 *Automation Maturity Progress States*

The maturity stages presented in Figure 2-2 align with the evolution cycles of automation and orchestration. Different sets of tools and methods can be used to automate and orchestrate the tasks at a given stage. This topic is covered in detail in the next few sections of this chapter.

You need to identify the current stage and identify the functional requirements to move to the next stage. As an example, if an organization is at a manual stage, you cannot directly move to the automated or self-driving stages because there are many dependencies. One such example is that moving to the automated stage requires integration with multiple systems and orchestration engines using application programming interfaces (APIs). A logical step between manual and automated stages will be the use of central controllers that could expose the rich set of APIs that can be used to integrate with various systems. The semi-automated stage will fill this gap because you will introduce API-rich domain controllers in this phase.

Please note this is merely an approach that is presented in this section. You can use any format or method to track the logical progression toward the next maturity level of automation and security, but the end goal remains the same: You need to target a self-healing and self-protecting network.

Building Blocks of Secure Automation

Secure automation relies on strong security principles that enable smooth and resilient operations. It focuses on protecting APIs to block unauthorized access, conducting ongoing security tests to find vulnerabilities, and incorporating advanced threat detection and response systems. By merging extended detection and response (XDR), security information and event management (SIEM), and security orchestration, automation, and response (SOAR), organizations can improve visibility, simplify incident response, and proactively address risks. These components collaborate to form a dynamic, self-defending security framework that adjusts to changing threats.

Secure Automation with API Security

Automation and orchestration depend heavily on the API interaction between different components. APIs are a common way to integrate between different on-premises and cloud-based network components. A typical SDN network controller interfaces with other components using different sets of APIs. The four common API types supported on Cisco Network Controllers such as Cisco Catalyst Center are

- **Northbound APIs:** These APIs allow higher-level applications such as management and orchestration tools to interact with the underlying infrastructure via the domain controller. These APIs enable programmatic access to the network services, configuration, and state information. Cisco domain controllers like Cisco Catalyst Center and ACI support northbound APIs.

- **Southbound APIs:** These APIs allow domain controllers like Cisco Catalyst Center to connect, configure, and manage network devices like routers and switches. These APIs act as a bridge between the controller and the control plane of the network device. These APIs are used for different functions including (but not limited to) device management, telemetry and monitoring, policy enforcement, and provisioning. Support to manage multivendor devices could be added with customer APIs/SDK packs.

- **Eastbound APIs:** These APIs are used to publish notifications that enable third-party applications to act on system or operational notifications. As an example, when a device fails a compliance check, an eastbound API notification can be sent to verify any software dependency and trigger an update if required.

- **Westbound APIs:** These APIs allow integration with information technology service management (ITSM), IP address management (IPAM), and reporting systems. These bidirectional APIs allow the exchange of contextual information between domain controllers and external IT systems.

In general, east- and westbound APIs can be combined into an east-west API interface, covering all operations through a single interface.

A lot of information can be shared using the APIs. It is important to apply the concept of zero trust to API security. It means securing the applications against importing vulnerabilities. Unsecured APIs can pose significant risks to a zero trust model. Vulnerabilities such as insufficient authentication, input validation flaws leading to injection attacks, and lack of encryption can expose sensitive data and systems. APIs relying on unverified third-party libraries or poorly maintained code can introduce hidden threats, while insufficient code scanning may leave security gaps undetected. In a 2019 study, Gartner (https://www.gartner.com/en/documents/3956746) identified API security as a new attack vector.

Open Worldwide Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software and has identified the following challenges with API security:

- **Weak Authentication Support:** When authentication mechanisms are implemented incorrectly, it opens the door for attackers to compromise authentication tokens or exploit flaws in the implementation. This can result in the temporary or permanent hijacking of users' identities. An API is vulnerable if it has the following (but not limited to) characteristics:

  - Uses weak encryption keys
  - Permits weak passwords
  - Sends sensitive tokens and passwords in the URL
  - Does not validate tokens

  Always ensure that all possible flows authenticate to the API. Please note OAuth is not the authentication. API keys should not be used for user authentication and should be restricted to the authentication of API clients only. Ensure you have created flows for reauthentication for sensitive operations such as changing the security policy in network firewall.

- **Broken Object-Level Authorization**: It is a security issue when an application fails to verify that a user has the necessary permission to perform actions on a specific object. If this situation is not implemented properly, attackers can exploit this flow and gain access to unauthorized data. This situation may result in network security issues. As an example, if the API does not verify the requesting user permission to access or modify the specific security policy ID, an attacker can make changes to firewall rules or security policies resulting in a security breach.

Make sure to implement authorization checks to ensure that users can only access or modify objects (policies in the case of the earlier example) that they are authorized to manage. Also, use indirect references or unique tokens to represent every object, making it difficult for attackers to guess valid IDs.

- **Broken Function-Level Authorization:** This issue refers to the failure to enforce correct permissions for accessing the specific network functions or operations. This situation might result in unauthorized users executing privileged operations. An example of this could be the automated script that performs critical operations like firewall rule updating is accessible without proper role-based authorization.

  To mitigate these risks, it is crucial to implement robust role-based access control, use strict authorization checks, maintain detailed logs, and conduct regular security audits.

- **Lack of Rate Limiting:** This issue refers to the situation where there is no restriction on the number of API requests a client can make within a specified timeframe. This can result in several security and performance issues, such as denial-of-service (DoS) attacks or resource exhaustion. It is also important to put in rate-limiting checks for fair access.

  You can use techniques such as token buckets or deploy an API gateway to handle rate limiting, authentication, and other security measures. Client-side handling like exponential backoff could also be implemented. It is also important to employ different test strategies for API security.

Dynamic Application Security Testing (DAST)

Dynamic application security testing is a security testing methodology that evaluates an application in its running state. In this approach, external attacks are simulated. There are various tools available in the market to help with DAST.

DAST can be adopted for API security by applying its concept to cross-site scripting (XSS) tests or cross-site request forgery (CSRF).

1. **Cross-Site Scripting (XSS) Tests:** These tests aim at identifying flaws that occur when an attacker injects malicious scripts into the content and an unaware user executes it later. The DAST approach can be applied to XSS tests in the following manner:

   - **Simulating Injection:** You can simulate the injection of various types of scripts into the API parameters and analyze the response to identify whether the scripts are executed.

   - **Response Analysis:** Tools need to examine the API responses in detail for any indication of rogue scripts being executed.

   - **Reporting:** You can identify and report endpoints that are vulnerable to XSS attacks and notify NetDevOps engineers to apply proper input and output validation.

2. **Cross-Site Forgery (CSRF) Tests:** These tests aim at identifying the vulnerabilities where an attacker tricks a user into performing an unwanted action on a network device management API where they are already authenticated. The DAST approach can be applied to CSRF tests in the following manner:

   - **Session Analysis:** DAST tools can analyze how the Application Under Test (AUT) handles different sessions and validate whether or not anti-CSRF tokens are used.

   - **Request Simulation:** Specific tools can then be used to simulate the unauthorized requests to the API, on behalf of authenticated users but without proper authorization.

- **Reporting:** The system can report the systems and identify the API endpoints lacking CSRF protection.

In addition to XSS and CSRF, DAST can be applied as a comprehensive security assessment of the API as part of overall penetration testing.

Integrating XDR, SIEM, and SOAR

Security and automation are closely tied together. Integrating them starts with collecting the logs and data, analyzing them, and orchestrating steps to tackle new threats. SIEM, XDR, and SOAR working together can help organizations build a methodological way to deal with new and existing threats. Before getting into the integration of these three complementing cybersecurity approaches, let’s look at the individual components:

- **Security Information and Event Management (SIEM):** SIEM is an approach or solution that allows the aggregation of logs and data from various disparate systems in an IT network. This data is collected from servers, network devices, security tools, and the like. This collected data is then aggregated and analyzed, and alerts are generated for any potential attacks.
- **Extended Detection and Response (XDR):** Whereas the primary function of SIEM is centralized log management and real-time monitoring, XDR is more focused on the use of advanced AI/ML-based contextual analysis and correlation of the data, to enhance the security threat detection and response.
- **Security, Orchestration, Automation, and Response (SOAR):** These platforms aggregate, correlate, and analyze results. They also create an automated incident ticket with relevant data and allow you to create response workflows to take action.

As you might have noticed, there are some overlaps between these technologies. However, as the systems are evolving, AI/ML-based analysis is taking precedence. The analysis part of SIEM and XDR might merge in the future.

By combining SIEM, XDR, and SOAR technologies, you can create a strong and automatic security response system. SIEM helps by collecting and analyzing security data from the network in real time to detect any threats. XDR makes detection better by bringing together data from many sources, like computers, networks, and the cloud, to give a complete view of the threats. SOAR helps by automating repetitive tasks and making the response faster and more consistent. Together, these technologies help to find threats early, respond quickly, and reduce the damage caused by security problems.

To illustrate how SIEM, XDR, and SOAR work together, let's look at the typical attack cycle shown in Figure 2-3.

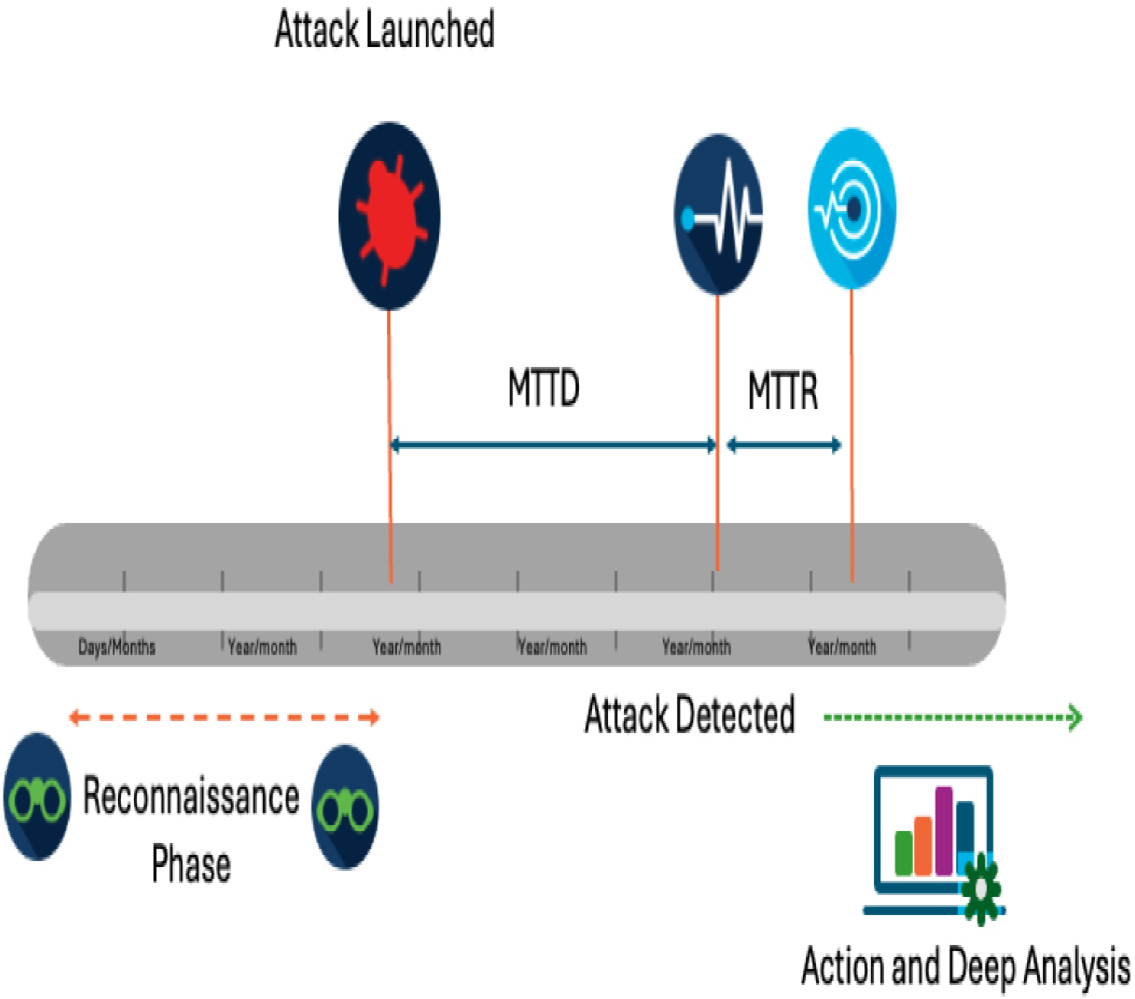


Figure 2-3 *Typical Attack Cycle*

Before any attack, an attacker spends time in the reconnaissance phase understanding and analyzing the system for potential attacks they can launch. The attack is then launched on a specific day, and it remains undetected for a certain amount of time. In some cases, it may go unnoticed for a very long time, like days, weeks, or even months. This time between the actual attack launch and attack detection is known as the mean time to detect (MTTD), and then action taken to resolve the issue temporarily or permanently is represented as the mean time to resolve (MTTR). Organizations then work on the root cause and deploy solutions to protect against such attacks in the future.

In the previous scenario, if the attack is new, then existing automation or security policies will not be able to detect and contain it. Also, you cannot automate threat containment in this case because you have not seen it before and are not sure of what action can be taken. In this case, you can use the combined strengths of SIEM, XDR, and SOAR to reduce the MTTR for any new attack. Once the deep analysis is done and threat containment steps are confirmed, you can make it part of standard threat containment and automation.

To understand this situation better, let’s consider a hypothetical scenario where a corporate firewall is under attack by an attacker and the attack type is not documented. As a first step, logs and information collected from the firewall (or multiple firewall instances) are sent to SIEM. This information is then sent to the XDR for deep analysis and correlation, which identifies the abnormal behaviors. XDR opens a new investigation case with SOAR with all collected artifacts so that security analysts can review the incident. The security analysts then use the concept of *dynamic playbooks* (A series of steps are taken to control a threat, where each step dynamically adjusts based on the outcome of the previous one) to create a series of actions to be taken to resolve the threat. SOAR systems have an easy interface to create these playbooks. You can even orchestrate the actions using the SOAR. Once defined, the action can be automated for future repeat incidents. Figure 2-4 shows the relationship between these three technologies.

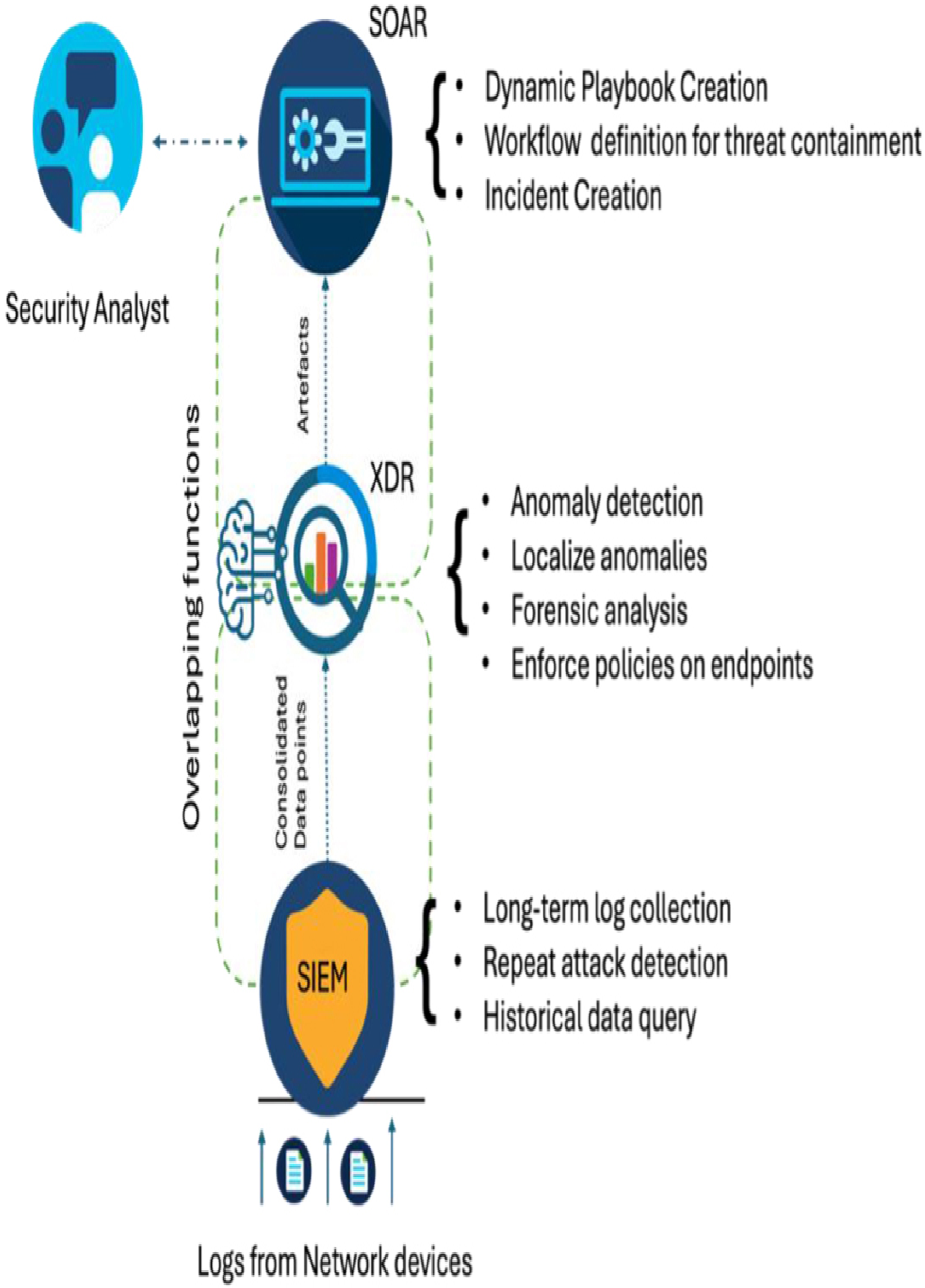


Figure 2-4 *SIEM, XDR, and SOAR Working Together*

Common Automation Practices and Tools

DevOps is a common software development practice that bridges the gap between the development and operational teams. DevOps, when combined with network practices, is known as *NetDevOps*. It is a commonly used practice for network automation. IT networks are getting bigger with devices from multiple vendors. It is important to keep the entire system up-to-date in terms of both configuration and software code to protect it from any kind of network threats. This has led to a practice of keeping the network configuration as codes and deploying it to different devices like a software patch. Because security is of paramount importance in any infrastructure, it is integrated early in the development and deployment cycle of NetDevOps. As you will learn later in this chapter, NetDevOps allows the identification and mitigation of security vulnerabilities to be detected early in the pre-production stage, only reducing the risk of major issues in the production environment. NetDevOps fits well with the need for modern IT networks because of the following:

1. Smaller but very frequent changes need to be applied to IT network systems. These changes could be due to evolving use cases resulting in configuration changes, or the need to mitigate threats due to a software defect or a new feature. These changes need to be tracked in a version-controlled manner.

2. NetDevOps allows rolling out updates and changes more reliably with the ability to roll back to the previous state using automation.

3. The NetDevOps approach avoids network disruption caused by bad configurations because this approach covers the validation at multiple phases.

4. You can easily integrate security practices into the CI/CD/CT pipeline, ensuring network changes are secure and compliant.

Key concepts of NetDevOps include

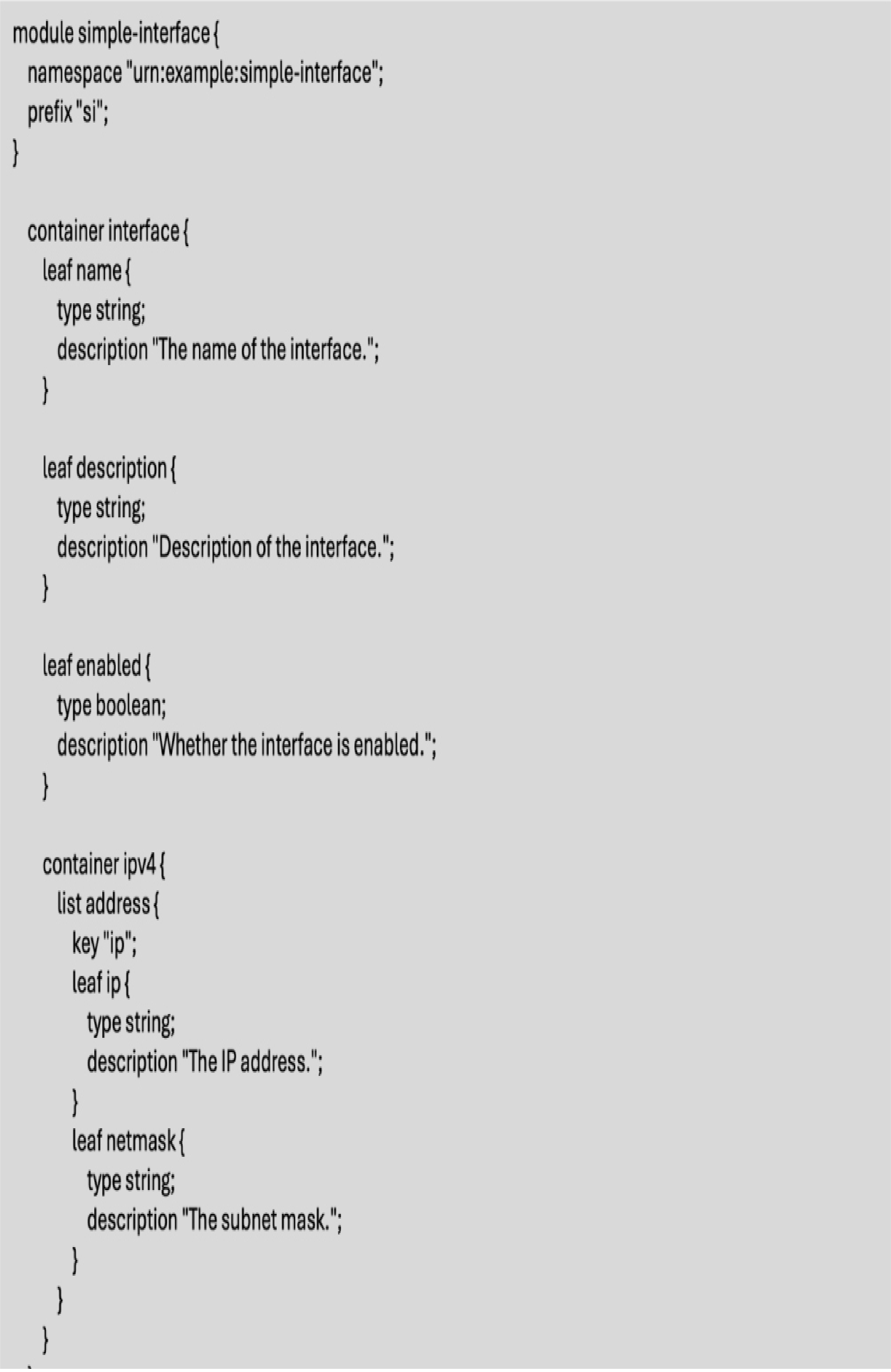
- Infrastructure as Code (IaC)

- Continuous Integration/Continuous Development/Continuous Testing (CI/CD/CT)

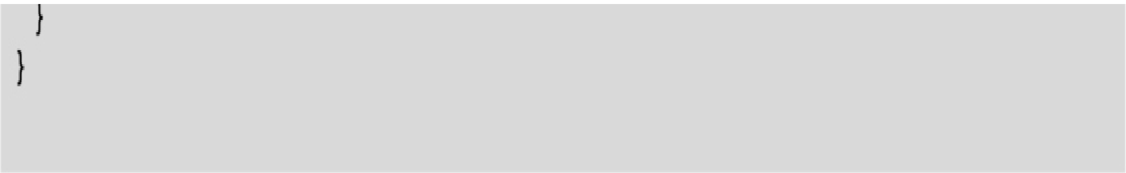
IaC and CI/CD/CT are key to modern network automation. IaC turns network settings into code, making them easier to manage and update. When combined with CI/CD, these codes can be automatically tested and deployed. This means network changes happen quickly and correctly, reducing mistakes and improving efficiency. Together, IaC and CI/CD/CT make networks more reliable and easier to manage.

Now let's examine in detail the concepts of IaC and CI/CD/CT. Infrastructure as Code is a practice that involves managing and provisioning network infrastructure through code rather than through manual processes. It brings software engineering principles, like version control and continuous integration, to network management. Think of it as writing programs or scripts to configure network devices but in a more process-oriented manner to reduce the risk of errors and make configurations consistent. IaC allows a standards-based programmatic method of writing the configurations and gathering the telemetry information from the devices. The following steps are typically involved in managing the infrastructure using IaC:

- **Data Modeling:** This step refers to defining the structure and constraints for a specific configuration. YANG is a common data modeling method that is written using XML, meaning an instance of the YANG model is an XML file. YANG models are specific to the network devices and are provided by the device manufacturers; for example, Cisco provides YANG models for its IOS, IOS-XE, and IOX-XR platforms. Standard bodies like the IEEE and OETF also define YANG models for common protocols and technologies. Figure 2-5 shows a YANG model for defining the interface on a Cisco IOS-XE device.



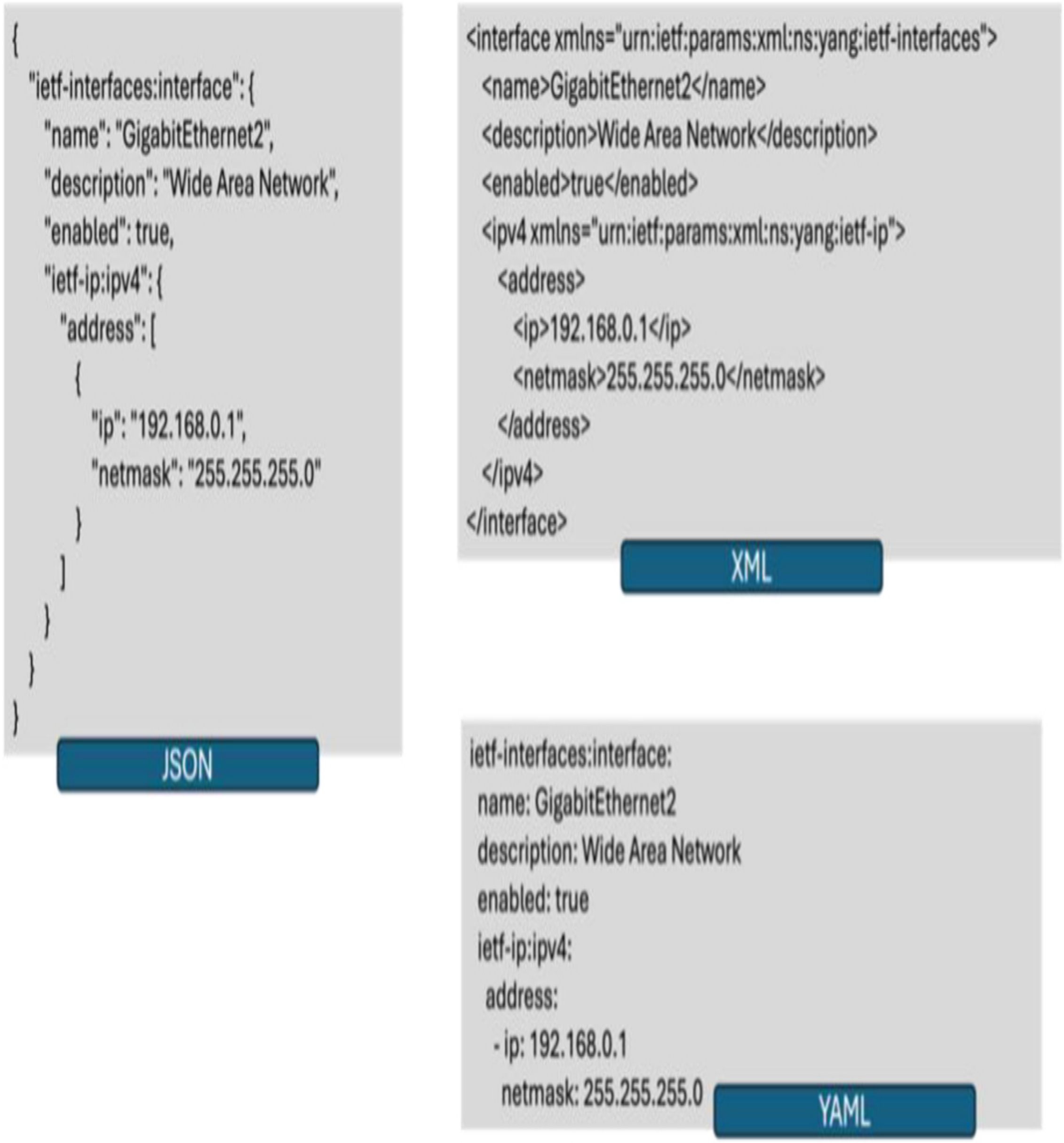
```
module simple-interface {
  namespace "urn:example:simple-interface";
  prefix "si";
}

  container interface {
    leaf name {
      type string;
      description "The name of the interface.";
    }

    leaf description {
      type string;
      description "Description of the interface.";
    }

    leaf enabled {
      type boolean;
      description "Whether the interface is enabled.";
    }

    container ipv4 {
      list address {
        key "ip";
        leaf ip {
          type string;
          description "The IP address.";
        }
        leaf netmask {
          type string;
          description "The subnet mask.";
        }
      }
    }
  }
```



```
  }
}
```

Figure 2-5 *YANG Model for Interface Definition*

- **Data Serialization Formats:** Serialization is the process of converting a data object, which is a combination of code and data, into a stream of bits and bytes. The most common data serialization techniques used for IaC are XML, JSON, and YAML. All these formats are human-readable. You will use these formats for defining the configuration for infrastructure components. In simple terms, you are adding actual data for the YANG models. You can define nested and complex data relationships using these data serialization formats. Figure 2-6 shows the definition of the IPv4 interface in all three formats. You will use the right serialization methods based on the network management protocol you intend to use (NETCONF or RESTCONF). YANG models are often serialized into XML for use with NETCONF or into JSON for use with RESTCONF.



```
{
  "ietf-interfaces:interface": {
    "name": "GigabitEthernet2",
    "description": "Wide Area Network",
    "enabled": true,
    "ietf-ip:ipv4": {
      "address": [
        {
          "ip": "192.168.0.1",
          "netmask": "255.255.255.0"
        }
      ]
    }
  }
}
```

```
<interface xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
  <name>GigabitEthernet2</name>
  <description>Wide Area Network</description>
  <enabled>true</enabled>
  <ipv4 xmlns="urn:ietf:params:xml:ns:yang:ietf-ip">
    <address>
      <ip>192.168.0.1</ip>
      <netmask>255.255.255.0</netmask>
    </address>
  </ipv4>
</interface>
```

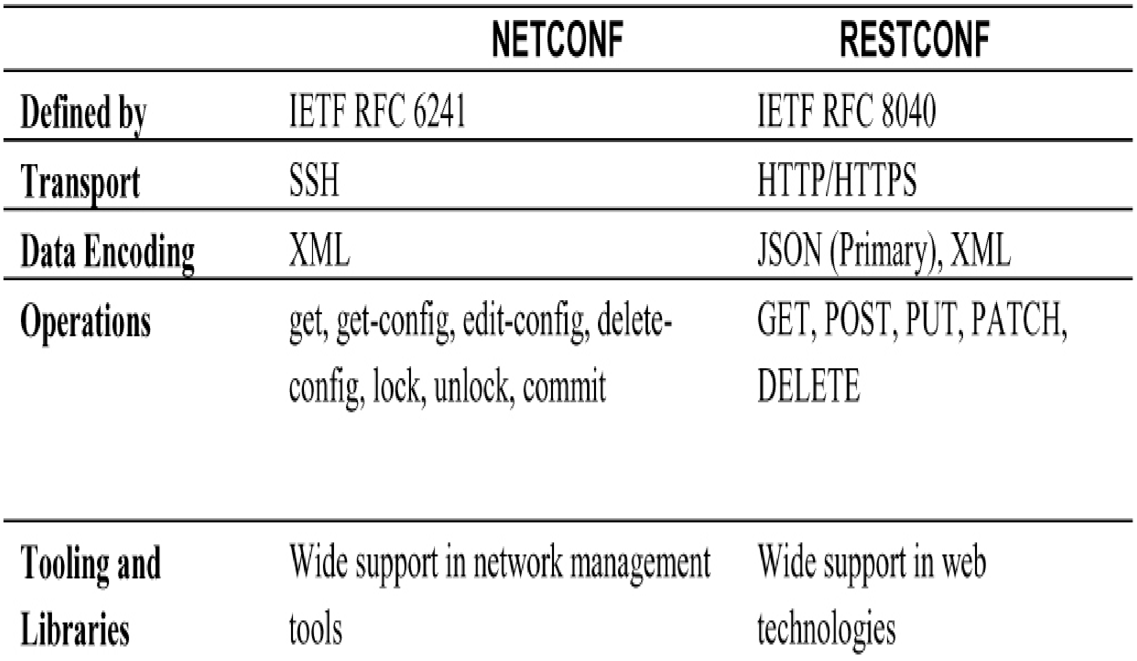
```
ietf-interfaces:interface:
  name: GigabitEthernet2
  description: Wide Area Network
  enabled: true
  ietf-ip:ipv4:
    address:
      - ip: 192.168.0.1
        netmask: 255.255.255.0
```

Figure 2-6 *IPv4 Interface Definition in JSON, XML, and YAML*

- **Network Management Protocols:** Once you have defined your intent in the form of a data serialization format, you need to find a way to send this information to your network devices. This is where the two common protocols NETCONF and RESTCONF come to your rescue. NETCONF (Network Configuration Protocol) and RESTCONF (RESTful Configuration Protocol) are network management protocols designed for configuring network devices, retrieving operational data, and managing network functions. They are both used for network automation and programmability but differ in their approaches and underlying technologies. In the example of network interface configuration, you will use a scripting language like Python to send the network configuration to the device using NETCONF.

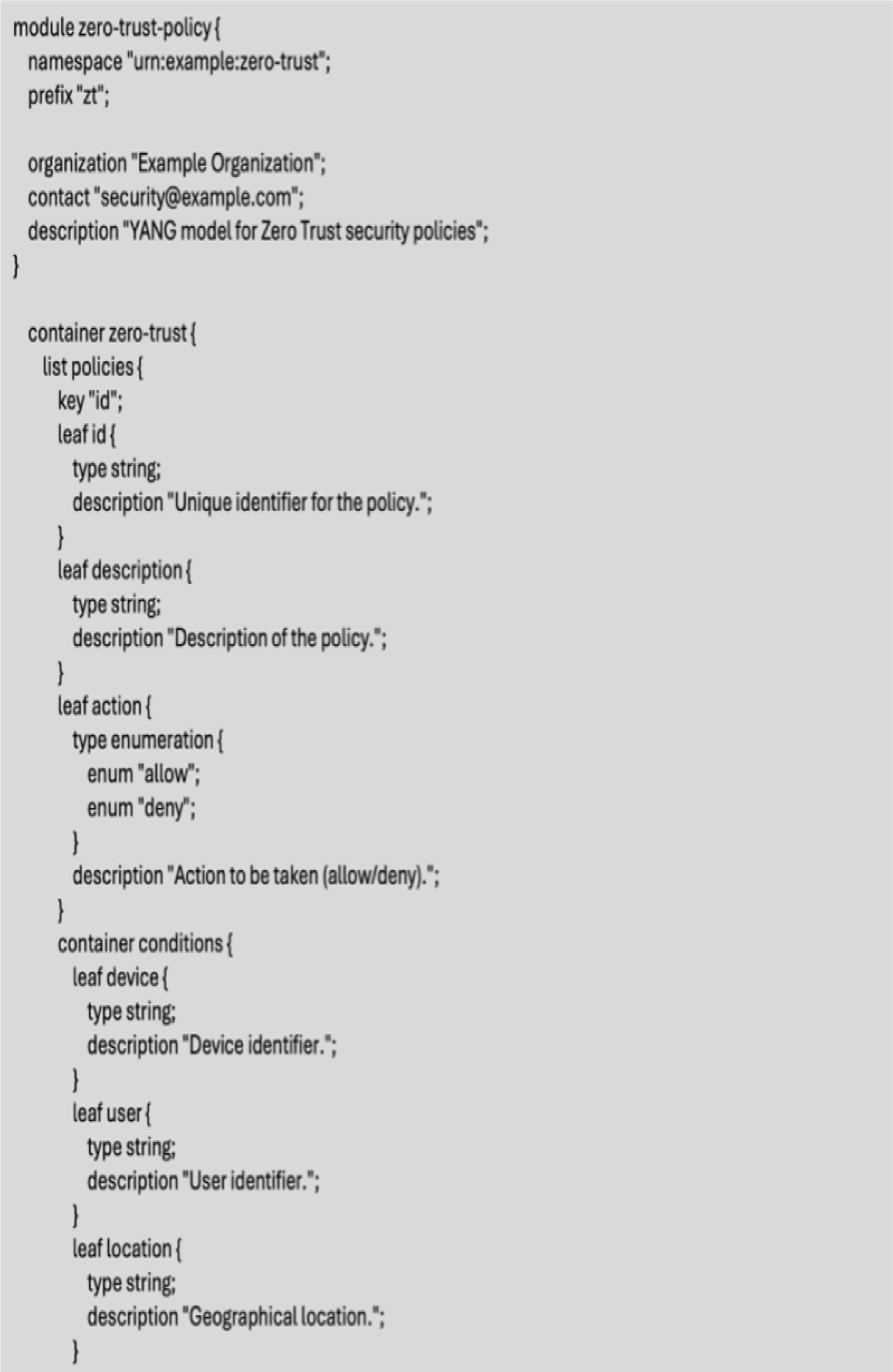
Table 2-1 compares NETCONF and RESTCONF.

Table 2.1 *Comparison of NETCONF and RESTCONF*

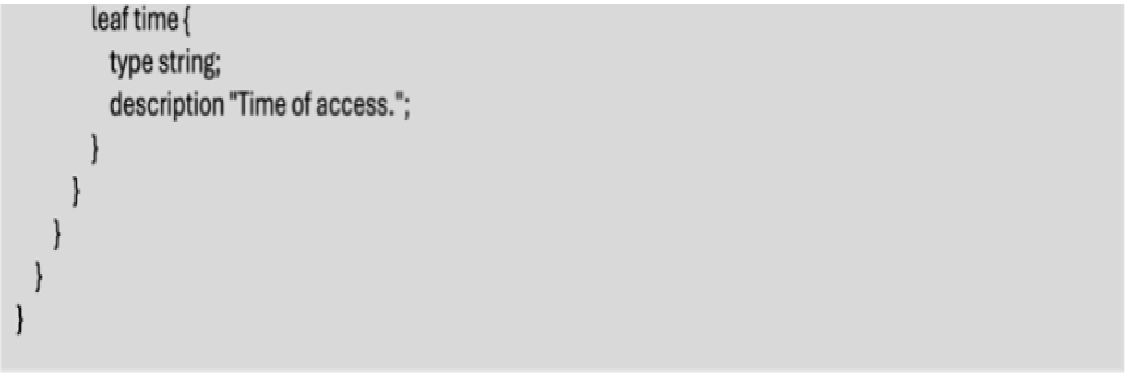


	NETCONF	RESTCONF
Defined by	IETF RFC 6241	IETF RFC 8040
Transport	SSH	HTTP/HTTPS
Data Encoding	XML	JSON (Primary), XML
Operations	get, get-config, edit-config, delete-config, lock, unlock, commit	GET, POST, PUT, PATCH, DELETE
Tooling and Libraries	Wide support in network management tools	Wide support in web technologies

From a zero trust perspective, you can use the approach described earlier where zero trust policies can be defined as the YANG model. It is then serialized using XML or JSON and finally deployed using NETCONF or RESTCONF. Figure 2-7 shows an example of a YANG model for a zero trust policy.



```
module zero-trust-policy {
  namespace "urn:example:zero-trust";
  prefix "zt";

  organization "Example Organization";
  contact "security@example.com";
  description "YANG model for Zero Trust security policies";
}

  container zero-trust {
    list policies {
      key "id";
      leaf id {
        type string;
        description "Unique identifier for the policy.";
      }
      leaf description {
        type string;
        description "Description of the policy.";
      }
      leaf action {
        type enumeration {
          enum "allow";
          enum "deny";
        }
        description "Action to be taken (allow/deny).";
      }
      container conditions {
        leaf device {
          type string;
          description "Device identifier.";
        }
        leaf user {
          type string;
          description "User identifier.";
        }
        leaf location {
          type string;
          description "Geographical location.";
        }
```



```
          leaf time {
            type string;
            description "Time of access.";
          }
        }
      }
    }
}
```

Figure 2-7 *YANG File Example*

Orchestration Using Ansible

Ansible is an open-source tool to orchestrate network automation tasks. It is gaining popularity due to its ability to abstract the complexities of standard automation approaches. Ansible uses a simple, human-readable language (YAML) to define automation tasks in *playbooks*. These playbooks describe the desired state of the system and the steps needed to achieve it. Enterprise security policies can be defined as these playbooks. Ansible then executes these tasks on the target systems, known as *nodes*, in an agentless manner, using SSH or APIs for communication. The typical steps required to do network configuration using Ansible are as follows:

1. Install the Ansible network modules—provided by the device manufacturer.
2. Create the Ansible playbook to configure the device.
3. Create an inventory file with the network devices.
4. Run the playbook.

As you may notice from these steps, Ansible abstracts much of the complexity, so network admins can focus on desired outcomes rather than low-level details. The YANG, XML, and NETCONF approach provides more granular control but at the cost of increased complexity. An easy way to orchestrate your network is to push changes to network controllers like Cisco Catalyst Center via Ansible. While systems like Cisco Catalyst Center support native automation capabilities, a network admin still has to configure and execute multiple steps for feeding that intent in the system.

Ansible can automate all the steps using a single playbook example creation of multiple new site hierarchies or software image upgrades for devices in a specific site. Cisco actively develops new Ansible modules to support new workflows. You also can interact with APIs via software development kits (SDKs). An SDK abstracts the lower-level details of API interactions, such as constructing URLs, handling authentication, and managing sessions. This reduces the amount of code required (repeated code) in Ansible modules, making it easier and quicker to implement complex workflows. Figure 2-8 shows the interaction with Cisco Catalyst Center and Ansible Playbooks.

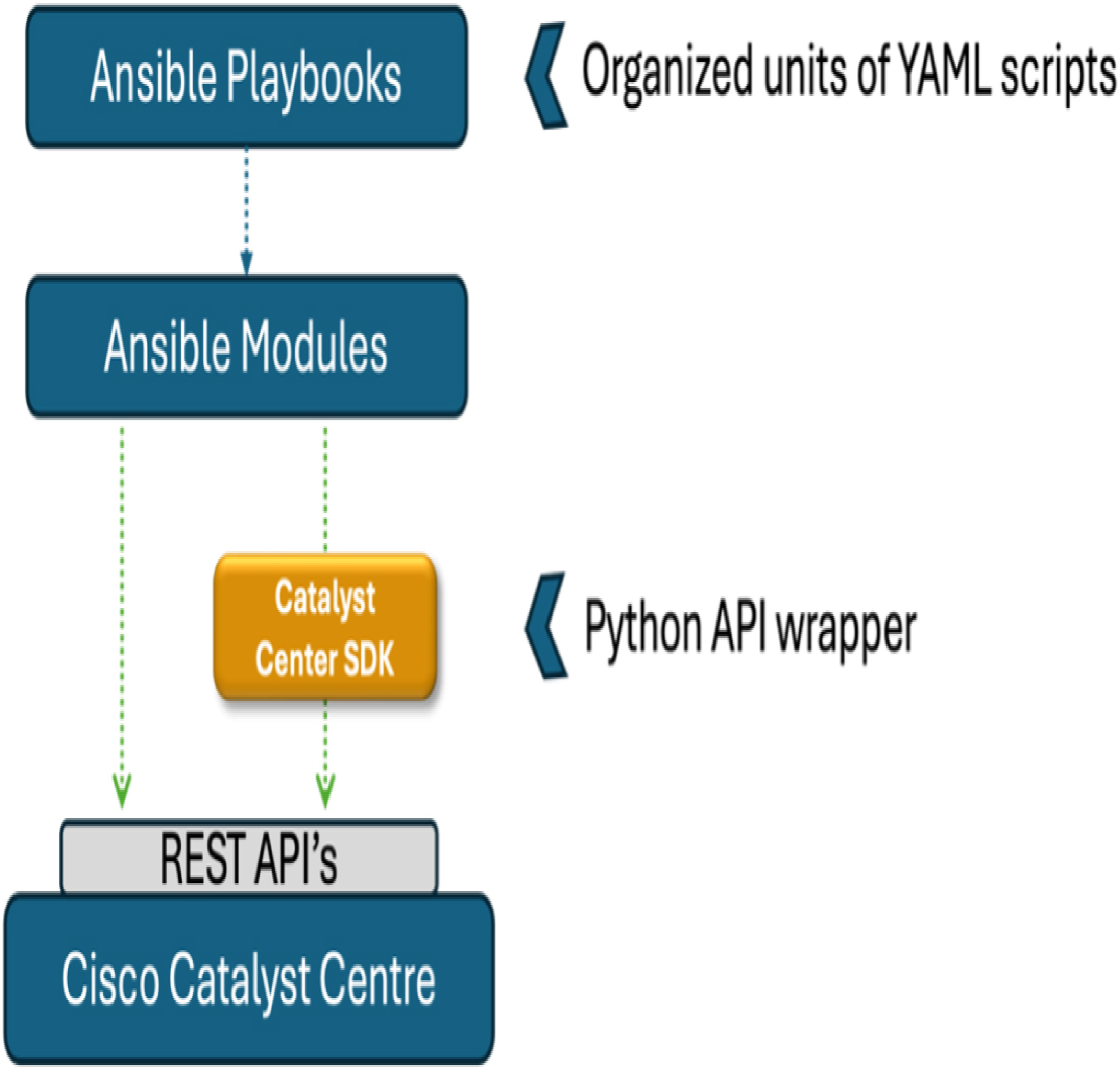


Figure 2-8 *Ansible-Based Orchestration for Cisco Catalyst Center*

Note

Refer to Cisco product documentation to get details on the Git repository to download Ansible modules for different products.

Note

IaC is covered in detail as part of Chapter 23, “Infrastructure as Code (IAC).”

Orchestration Using Terraform

Terraform is an open-source IaC tool that allows for the automation, management, and provisioning of infrastructure in a declarative way. At its core, it uses components like providers, which serve as handy plug-ins to connect with specific APIs or platforms, and the state, which keeps track of the infrastructure's current condition. A provider is responsible for understanding API interactions and exposing resources. Providers generally are IaaS (e.g., AWS, GCP), PaaS, or SaaS. However, they can be used for custom solutions like Cisco ACI.

You can orchestrate different Cisco solutions using Terraform. Let’s consider an example of Cisco ACI: Terraform's ACI provider offers an effective way to work with the Application Policy Infrastructure Controller (APIC). With this provider, you can effortlessly define and manage key ACI components such as tenants, application profiles, endpoint groups (EPGs), bridge domains, and contracts, all by using simple declarative configuration files written in HashiCorp Configuration Language (HCL). The Terraform ACI provider connects directly with the APIC API, making it really easy to streamline provisioning, updates, and lifecycle management of ACI constructs.

CI/CD/CT

Continuous integration, continuous delivery/deployment, and continuous testing are essential practices that help automate and streamline the lifecycle of network configuration, management, and infrastructure changes. Let’s look at the individual components of CI/CD/CT

- **Continuous Integration (CI):** CI involves automatic integration of the code changes to the central repository. It can be applied to the configuration scripts such as those written in YAML or Python. As part of NetDevOps, automated tests are performed to check for any syntax errors, compliance, or best practice. Changes to network configuration (including security policies) are managed via version control systems like Git. For example, when a network engineer pushes a change to the network configuration repository, the CI pipeline can validate if any new security policies introduced do not break the zero trust principles such as least privilege or microsegmentation.
- **Continuous Development (CD):** Continuous Delivery (CD) ensures that software is always in a deployable state with automated testing, while Continuous Deployment (CD) takes it further by automatically releasing updates into production without manual intervention. Typical stages included in this process are development, staging, and production. Multiple analysis and validation steps are included in each stage. Figure 2-9 shows the sample CI/CD/CT workflow. You will also notice that after the staging stage, Golden Config is generated, which can then be deployed in the production environment with confidence.

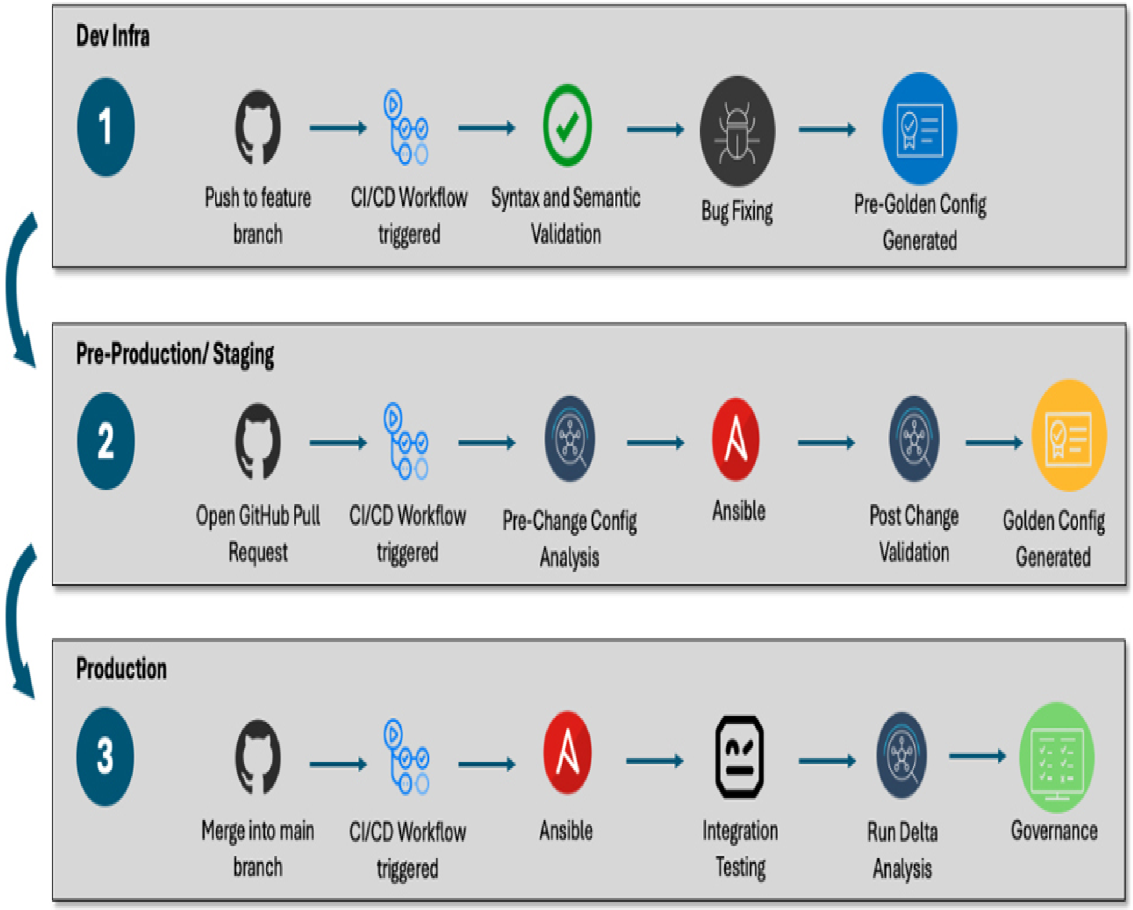


Figure 2-9 *CI/CD/CT Workflow*

- **Continuous Testing (CT):** CT involves integrating automated testing into the development pipeline to provide feedback on the quality and performance of the code throughout the development lifecycle. Testing modules can simulate different network attacks to test the effectiveness of new security controls or validate that access controls are correctly implemented as part of the new configuration set.

AI and Machine Learning with Automation

Artificial intelligence (AI) is a broad field of computer science that aims at creating computer systems that can perform tasks with similar intelligence to humans. There are several subfields within AI, such as *machine learning (ML)*, *neural networks*, and *large language models (LLMs)*. Figure 2-10 shows the relation between these AI subfields. Next, let's examine the basic differences and correlations between these technologies.

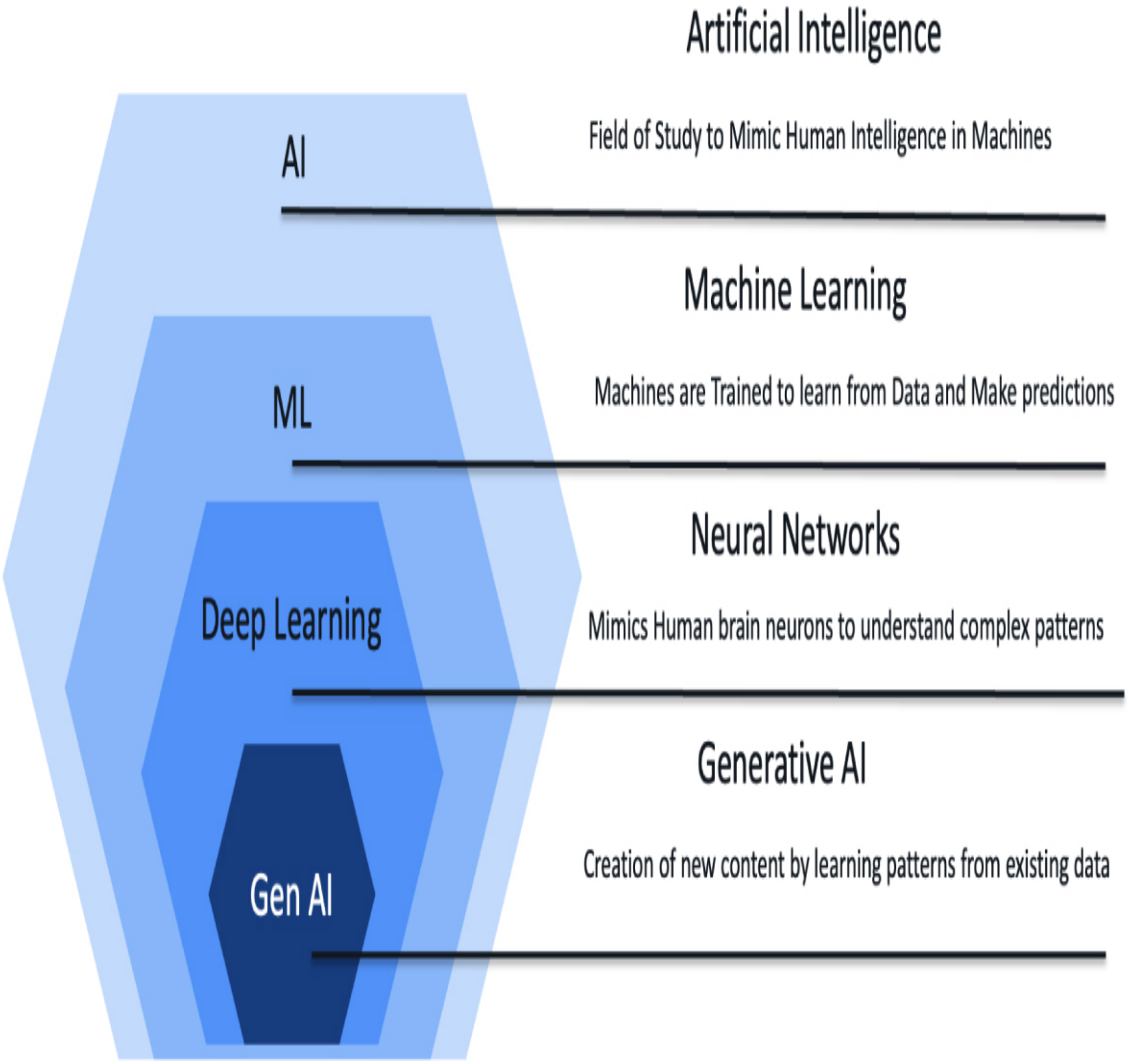

Artificial Intelligence
AI
Field of Study to Mimic Human Intelligence in Machines
Machine Learning
ML
Machines are Trained to learn from Data and Make predictions
Neural Networks
Deep Learning
Mimics Human brain neurons to understand complex patterns
Generative AI
Gen AI
Creation of new content by learning patterns from existing data

Figure 2-10 *Relation Between Common AI-Related Technologies*

Machine Learning (ML)

Machine learning is a subset of AI that focuses on developing algorithms that allow computer systems to learn and make predictions based on historical data and information. These algorithms improve their accuracy based on training data provided to them. A primary use case of ML is to help with predictions. You will see ML is used in different networking products to predict future behavior, such as congestion, wireless interference, and probability of network breaches based on the current state of these systems (like configurations, software code) and historical data on such incidents. The training data set is critical for ML algorithms to work optimally. You can train these models using different techniques; some of the common ones are as follows:

- **Supervised Learning:** In this method, ML algorithms are trained based on labeled data using classification and regression techniques. Classification allows models to put items in a specific category; for example, you can train an ML model to clarify email as spam" or "not spam." The model learns to classify the new emails based on parameters like sender, subject lines, and content. Regression is used to understand the relationship between dependent and independent variables. Applying regression to the spam email example, instead of just labeling emails as "spam" or "not spam," you can predict a continuous probability score that indicates the likelihood of an email being spam. This probability score can be used to prioritize emails for review or to set more dynamic thresholds for spam detection.
- **Unsupervised Learning:** In this method, ML algorithms are trained on unlabeled data where the model tries to find hidden patterns and intrinsic structures. One technique of unsupervised learning is clustering, which is a grouping and sorting of items with similar characteristics. For example, if you have a box of mixed LEGO pieces, you can group them based on similar shapes or color. Another example of applying this concept to network observability is detecting anomalous traffic patterns to help identify potential security breaches, misconfigurations, or other issues. Since anomalous traffic data is not labeled, unsupervised learning techniques like clustering can be used to identify these anomalies. Clustering objects in a simple manner by focusing on a primary feature, like the color of LEGO in the example, is known as dimensionality reduction.
- **Reinforcement Learning:** In this method, learning is motivated by rewards. Learning is through the trial-and-error method to maximize a reward. As an example, if you want to teach your dog to roll over, you give instructions and commands. When the dog performs the desired action, you give it a treat. Over time the dog learns that doing

the action or specific actions maximize the treat. Similarly, you can train the malicious traffic detection system and create a reward function to improve the trust score every time it detects the correct anomaly, to reduce false positives over time.

Neural Networks

A neural network is a type of machine learning model inspired by the human brain structure. It consists of interconnected nodes called neurons that are organized into different layers. It has one input, one output, and several hidden layers. Each connection between neurons is associated with the weight that adjusts based on new learning by the network.

- **Input Layer:** Features such as packet size, protocol type, source and destination IP addresses, and port numbers are used as input.
- **Hidden Layers:** These layers process the input data to extract patterns and relationships. For example, the network might learn that certain patterns of traffic are indicative of a distributed denial-of-service (DDoS) attack, while others are typical of normal operations.
- **Output Layer:** The network outputs a classification, such as "malicious" or "benign," based on the learned patterns.
- In a network environment, neural networks can be used for tasks like classifying network traffic. For instance, suppose you want to detect whether incoming network traffic is benign or malicious

Figure 2-11 shows a typical neural network with multiple input, output, and hidden layers. In the case of the previous example, because we intend to classify the traffic into the two categories malicious and benign, the number of output layers will be two in this case.

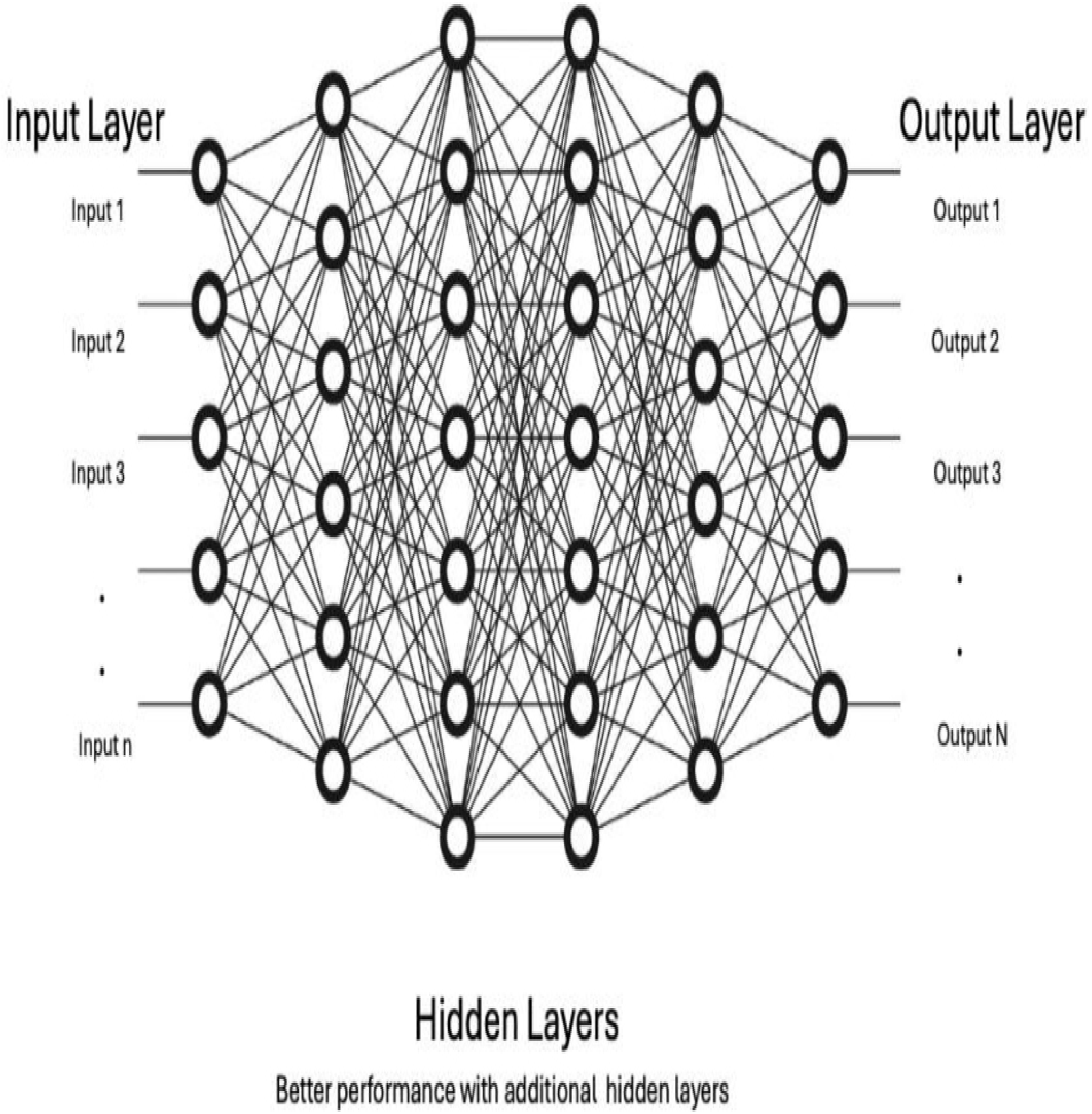


Figure 2-11 *Sample Neural Network with Multiple Hidden Layers*

Generative AI and LLMs

Generative AI (GenAI) is a broader field within artificial intelligence that encompasses various techniques and models, including but not limited to neural networks. It is designed to create new content or data that mimics the human-created material. It can generate new text, images, video, music, and so on.

Large language models (LLMs) are a type of GenAI that specifically focuses on understanding and generating human language. They are trained on vast amounts of data to learn patterns, grammar, and context. Once trained, LLMs can predict and generate text based on a given prompt. Because these models might not be trained on the latest data, it is often integrated with retrieval-augmented generation (RAG). RAG is primarily a method to add the latest domain-specific information to assist the GenAI. For example, if you want information on later cyber threats, LLMs may not return the best answer because they may have been trained a few months or years back. In such cases, RAG can search the latest threat intelligence reports, security advisories, and so on. The LLMS then use the latest information to produce more accurate answers.

GenAI and LLMs can significantly enhance network automation by providing intelligent solutions for network configuration, troubleshooting, and management. GenAI offers many opportunities to automate tasks. Some of the common use cases that actively use GenAI at the time of writing this chapter are as follows:

- **Issue Diagnosis:** LLMs can analyze the log files, error messages, and symptoms. Based on their diagnosis, they can suggest the troubleshooting points.
- **Debugging Network Scripts:** You can pass an error while running your scripts to LLMs to identify the issue, and they can do analysis to find the root cause and suggest steps to fix the issue.
- **Policy Generation:** GenAI can analyze the network configuration and compare it with established policies to identify any deviations and compliance.
- **Documentation:** You can use GenAI to automate the creation of documentation.
- **Adaptive Scripts:** You can use GenAI to write scripts that adapt based on evolving network conditions or requirements, guided by GenAI’s insights and recommendations.

Data Lakes

A *data lake* is a centralized repository that stores vast amounts of raw data in its native format until it is needed. As you might have noticed, data plays a critical role for all use cases related to AI/ML. An organization must have a data lake so that different models can be trained on that data. Even if the intention is not to use GenAI, a predictive analysis model requires this data set. Cisco Splunk is an example of a solution that provides a data lake capability with custom AI/ML-based analysis of the data.

As you learned at the beginning of this chapter, the end goal for any organization from a zero trust and automation perspective is to reach a self-driving state where threats are auto-contained and the network heals on its own. This could be achieved by employing different approaches, as covered in this chapter. Figure 2-12 covers a generic approach to have such a closed-loop automation system with zero trust across analytics, orchestration and automation, and infrastructure layers.

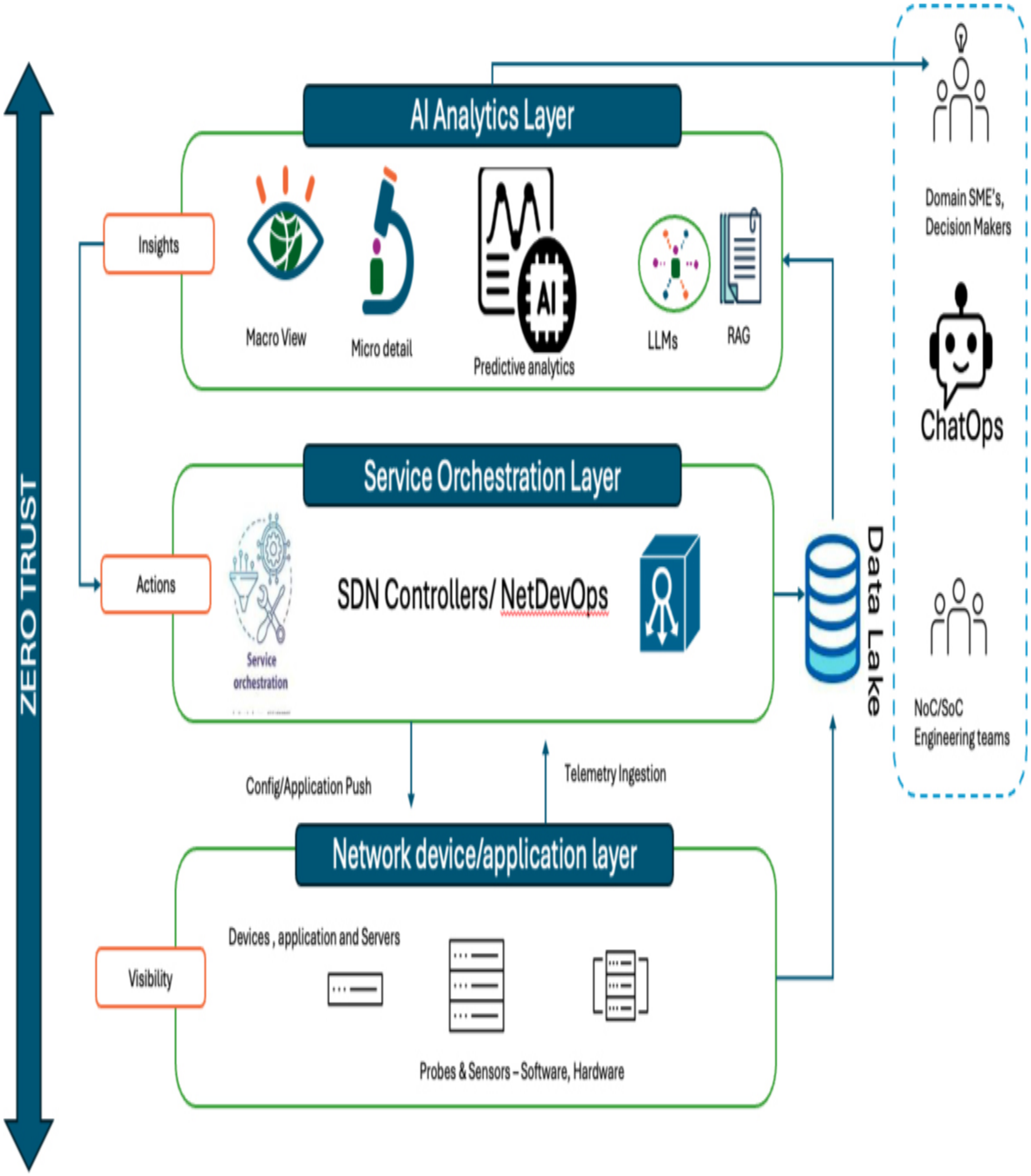


Figure 2-12 *Zero Trust Application Across Closed Loop Automation*

Zero trust must be applied across data lakes, automation, orchestration, and analytics to ensure consistent and comprehensive security.

- Data lakes provide a centralized view of all data, and applying zero trust ensures that access to sensitive information is tightly controlled and continuously monitored, reducing the risk of unauthorized access or data breaches.

- Automation in zero trust helps apply security policies consistently and without human error, ensuring that only trusted users and devices are granted access at all times, even in complex and dynamic environments.
- Orchestration integrates multiple systems, and applying zero trust across all of them ensures that security actions, such as threat detection or incident response, are coordinated and executed quickly, without delays that could lead to vulnerabilities.
- Analytics uses data to detect threats, and with zero trust, analytics can be used to continuously evaluate user behavior and network traffic, ensuring that only legitimate actions are allowed while identifying and blocking suspicious activities.

By applying zero trust across all these components, organizations can create a unified, automated, and proactive security environment, reducing risks and ensuring that every access request is thoroughly verified, regardless of its origin.

Summary

In this chapter, you learned how automation and orchestration play an important role in consistently applying configuration settings and policies that support the enterprise's zero trust strategy. These tools help ensure that the right security measures are deployed across the network and that they follow the zero trust principles.

However, policies by themselves do not automatically create a zero trust environment. The real goal is to implement and reflect a zero trust strategy by using the right configuration settings, creating appropriate security policies, and integrating them with automation and orchestration tools. This combination ensures that access is continuously verified, and security measures are consistently enforced across the organization.

Chapter 3. Zero Trust Network Deployment

In this chapter, you will learn about the following:

- Zero trust and functional pillars
- Elements of zero trust policy
- Tools and technologies for zero trust deployment
- Greenfield versus brownfield zero trust network deployment

In today's risky and uncertain times, organizations need to build a solid foundation for reliable and adaptable security operations. This could be achieved by building zero trust in their IT systems. The zero trust framework operates on the principle of "never trust, always verify." Unlike traditional security approaches that rely on a strong perimeter (like firewalls) to protect a trusted internal network, zero trust assumes that threats can originate both inside and outside the organization. Therefore, every request, user, or device must be authenticated and authorized, regardless of its location within or outside the network. It is important to understand the functional pillars of zero trust before any organization starts deploying it. You will need a variety of tools and platforms to deploy zero trust in your organization. Your zero trust approach must be able to perform the following four functions:

- **Establish Trust:** Ensure the system can verify each user and device connecting to the network.
- **Enforce Trust-Based Access:** The system should be able to enforce policies and grant access to the users and devices based on the trust level. The policies should be applied based on the principle of least privilege. The access mechanism should protect all network

resources, including applications and data, on-premises and in the cloud.

- **Verify Trust:** The system should be able to continuously verify the trust and dynamically adjust the access to the network. It should not be a time check and then access is allowed for a longer duration or to multiple applications.
- **Enforce Policies and Monitor:** The system should be able to react to changes in trust by analyzing and coordinating responses to potential incidents while gaining better visibility into suspicious activities related to trust levels. A continuous refinement of trust policies is required.

An organization needs to protect all its assets irrespective of their location. Some assets could be mobile-like users and devices, and others like applications, and stationary databases could either be on-premises or hosted in the cloud. The zero trust approach should consider various factors, such as user type, location, device type, data requested, and application accessed, to create robust dynamic policy-based access controls. Some of the common assets you want to protect in your organization include (but are not limited to)

- Users
- Devices
- Network
- Cloud
- Applications
- Data

A common question arises: Where to start, and how can you define the steps toward a zero trust framework adoption?

In the rest of this chapter, you will learn the iterative approach that could be adopted for securing trusted access for users and devices based on the four primary functional requirements of zero trust as covered in this section. We will first look at the key decision points to establish the overall zero trust strategy that includes establishing trust for users and devices, application access, and policy enforcement. Next, we will look at the common tools and technologies to help you deploy zero trust, and finally, we will look at the approach for zero trust adoption in greenfield and brownfield scenarios.

Elements of Zero Trust Strategy Definitions

In this section, we will look at the key considerations that you will need to formulate the zero trust deployment policy. These considerations include

- Establishing user trust
- Establishing device trust
- Defining application access policies
- Enforcing policies

Establishing Trust

A trust level represents the degree of confidence you have that a user or device is who or what they claim to be and that they are authorized to access specific resources. Unlike older models, where access is granted based on location (inside vs. outside the network), trust levels in zero trust are dynamic and can change based on several factors. Trust levels can be assigned to users and devices.

- **User Trust Level:** Based on the identity of the person accessing the network
- **Device Trust Level:** Based on the security status of the device accessing the network

The first functional requirement of zero trust is to establish trust for users and devices. This requirement could be divided into two categories: users and devices. Let’s look at both of them in detail.

User Trust Definition

The purpose of defining user trust is to verify the user's trust and enforce the principle of least privilege. You need to start evaluating what mechanisms and processes are there in your organization to authenticate and authorize the users before they can access the resources. But wait, which users? There could be different types of users having different privilege access. To start, do the following:

1. Identify the user types and roles (employees, including contractors, temporary users, and guests).
2. Associate the risk with each user type, based on their function and type of data they need to access.
3. Identify the assets that need to be accessed by these user groups.
4. Identify the location of assets each group needs to access—on-premises, cloud, DMZ, and so on.

Following these steps, you will get a matrix that shows the user group and their need to access, which type of information, and where that information is stored. As a next step, you need to define the trust level for each user type based on the authentication and authorization system in place or planned. You can start evaluating as follows:

- How does an employee need to be authenticated and authorized?
- How does a partner need to be authenticated and authorized?
- Can you deploy multifactor authentication (MFA)? If yes which applications are critical?
- Is biometric-based passwordless authentication required?
- How often is the MFA required, and what reauthentication triggers need to be deployed?
- Do you have a secure identity database for the full type or contract employees?

Once you have evaluated the responses to the questions, you will then need to devise a policy, plan, and then deploy the solution that allows strong authentication and authorization of the users before they can access any information. User trust can be verified using different methods and actions such as :

- **Multifactor Authentication (MFA):** Ensure users verify their identity through multiple factors (something they know, have, or are)—for instance, a password combined with a mobile authentication app or biometric verification.
- **Role-Based Access Control (RBAC):** Grant users access only based on their roles, ensuring they can access the minimum necessary resources.
- **Contextual Authentication:** Authenticate based on context (time of access, location, device type) to provide a more secure, tailored approach.
- **Just-in-Time Access:** Ensure that users get access only when they need it and that it's revoked immediately after their session ends, minimizing the risk of lingering access.
- **User Behavior Analytics (UBA):** Monitor user activity in real time, including access patterns and application usage. Anomalies (e.g., accessing data from an unusual location or time) trigger additional authentication steps or even block access.
- **Dynamic Risk Scoring:** Assign risk scores to users based on behavior, device posture, and external factors (e.g., suspicious login attempts). Trust levels adjust dynamically based on these scores.

For a greenfield scenario (described more fully later in this chapter), it is easy to plan, design, and implement first. Users are then onboarded to a well-defined and secure environment. But in a brownfield environment, you might face challenges such as users being hesitant to adopt new methods of authentication. Some workflows might need to be interrupted to adopt a new user identity verification system. To overcome some of these challenges, you must do the following:

- Prepare clear communication and messaging for the need for change.

- To avoid any hiccups due to technical glitches, ensure that all systems are tested. You can start the pilot with a small group and data set as identified in the matrix.

Device Trust Definition

Defining device trust means ensuring that a device is safe and authorized to access the network. You must never assume that any device is safe because it is inside the network or is company-owned. Especially with BYOD policies, employees are now allowed to bring their own devices to work. Such devices need to be secured and segmented properly. Ensuring devices accessing your network are in a healthy state is a critical step for zero trust deployment. You can start by doing the following:

- Evaluate the type of the company’s own assets, such as model, make, and operating systems.
- Rank devices based on the risk and data sensitivity.
- Create a list of technologies that can be used for device trust such as virtual private network (VPN), mobile device management (MDM), and certificates.
- Create a strategy based on supported devices and their capabilities such that each device can be continuously authenticated.

It is important to understand the difference between a company-owned asset and company-managed asset. A company-owned asset is owned, managed, and fully controlled by the organization, whereas a company-managed-only device typically includes BYOD devices that an employee can purchase but are managed by the company. This is typically done by enrolling devices in an MDM system that defines the security policy and configuration for the device such as certificate requirements or password lengths. You can even wipe an entire device remotely if an employee reports it as stolen. Access to further applications and resources could be constrained for such devices. Because managed devices are tracked, enrolled in configuration and patch management programs, and continuously monitored for security incidents, the trust level for such devices is always higher than for unmanaged personal devices. As such, your zero trust policy usually allows Internet-only access for unmanaged personal devices. We have discussed managed and unmanaged devices, but how do you bring a device under management? You can use a combination of solutions such as

- **Certificate-Based Trust:** In zero trust, every device must prove itself before accessing any resource. Using certificate-based trust, a device proves its identity by showing its digital certificate. Even if the device is already inside the network, it must keep proving its trustworthiness every time it tries to access new resources or data.

- **Device Health:** It is important to ensure that a device is healthy before it can be allowed on the company's network. Even a company-managed device with a certificate installed can get infected. It is important to use device posture as one of the device trust points. Validating if the endpoint security agent is installed to protect against the threats helps in establishing trust.

Device trust further strengthens the security because now an attacker needs not only a valid user credential but also a trusted device to launch any attack. A combination of trusted users with trusted devices is commonly used as part of the zero trust policy definition to provide different levels of access. Typical components you will require for device trust include

- **PKI Infrastructure:** PKI (Public Key Infrastructure) is a framework that manages digital certificates and encryption keys to enable secure communication and authentication over networks. In zero trust PKI is used to issue device certificates. These issued certificates will be used for identifying all company-managed assets, including company-owned or BYOD assets.

- **Security Agents:** A variety of tools can be installed on company-owned assets, such as endpoint agents, VPN, VPNless remote access, and device posture agents. For BYOD, an MDM solution could be used.

- **Central Inventory:** It is important to have a central inventory of all devices in your network with proper asset tags, owners, type of access required, and so on.

One common challenge with device trust is the hesitation of users to install any agent or certificate on their devices due to privacy concerns. This requires careful validation based on the local laws, so the organization must use a solution to verify the trust without violating privacy. As an example, monitoring the website that a user visits on a BYOD could be considered a privacy issue, while suggesting a stronger login password with a specific key length could be considered a genuine safety measure.

With user and device trust policies established, you now know who can access the network using which types of devices. We have covered a high-level approach, but you need to go multiple levels down to establish detailed policies, tools, procedures, and ways to track the user and device trust. Once you have done this, the next step is to enable access to the application by framing policies that a combination of users and devices can access.

Trust Score Calculation

Trust levels can be represented as scores or risk levels based on multiple factors. Here's a simple breakdown of how they might be calculated.

Each factor gets a score, and the total score determines the user/device trust level:

- Base Trust for User Credentials (e.g., MFA or SSO): 50 points
- Device Security (e.g., fully patched, antivirus running): 30 points
- Geolocation (known/unknown location): 10 points
- Time of Access (normal/abnormal): 10 points

If all conditions are met, the trust score would be 100. A lower score (below a set threshold, e.g., 70) might trigger additional security actions, like reauthentication or restricted access

Instead of adding points, you can subtract trust based on risk indicators. For instance:

- Base Trust: Start at 70.
- Subtract 40 if the user uses only a password (no MFA).

- Subtract 20 if the device is missing recent patches.
- Subtract 10 for unrecognized geolocation.

Once a trust score or risk level is assigned, the system decides how much access to grant and whether any additional actions are needed:

- **High Trust Score (90–100):** Full access to the system, no extra verification needed.
- **Medium Trust Score (70–89):** Limited access, or additional verification required like MFA.
- **Low Trust Score (Below 70):** Deny access or allow access only to less-sensitive resources.

Defining Application Access Policies

The entire idea of establishing trust for users and devices is to allow them access to business applications in a secure way. This ensures that even after the user and device are verified, they get access only to specific applications they need. One of the important considerations you need to make is where the application and data are hosted, such as public or private cloud, on-premises, or SaaS applications. You will need to devise the policies for different data sets and applications based on how often they are accessed, keeping in mind the user experience you want to provide. Following are some of the decision points for creating policy around application access:

- A comprehensive list of applications used in the organization, both internal and external, should include cloud-based, on-premises, and hybrid applications.
- It is also important to identify the application dependencies like interactions with databases, other services, and APIs. This information will come in handy while building the microsegmentation.
- What user types need to access the application? We covered this step from a user authentication perspective earlier in the user trust definition, but now you need to map the application in detail, along

with the frequency at which different user groups typically access any applications, and rank the risk of the applications.

- Policies also need to consider device type and location of user groups concerning the application accessed. Do you want to reauthorize or have stricter controls when users try to access the service from a location that is different from their regular location? If your entire workforce is mobile, then your strategy needs to include the fact that random location is normal behavior. In such a case, frequent location-based reauthorization may result in poor user experience. You may want to club the common application sets and allow access based on single sign-on (SSO).
- The SSO-based approach is essential because users often forget passwords for multiple applications. When many applications are involved, users may resort to using the same password for all. Here, SSO provides a centralized enforcement point, allowing the establishment of policies on password length and change frequency. It is advisable to implement multifactor authentication with biometrics alongside SSO.

Access policies should align with user and device trust levels. Trust depends on contextual factors like location, time, and behavior, necessitating continuous monitoring even for authorized access. For example, if a terminated employee quickly downloads multiple files, this could indicate potential malicious intent. Such behavior should trigger alerts or adjust access per established protocols. This strict approach reflects a zero trust framework, which emphasizes that no one is inherently trusted. Ongoing supervision and adaptive policies are vital to enforce zero trust principles.

Macro- and microsegmentation form the basis of the zero trust network access, as you will learn later in this book. From an application perspective, it is also critical that segmentation considers the following constructs:

- **Application-Level Segmentation:** Isolate applications from one another so that a compromise in one doesn’t lead to a breach in others. This involves defining granular policies for inter-application communication and blocking unnecessary pathways.

- **Data and API Segmentation:** Segregate databases, APIs, and other backend services from applications and users that don't require direct access.
- **Environment Segmentation:** Separate development, testing, and production environments to ensure that an issue or breach in one environment doesn't spread to others.

It is also important to think about securing the data in motion. When the authorized user groups access specific applications, it must be secure. Following are some of the approaches you may consider:

- **Transport Layer Security (TLS):** Ensure that all traffic between users and applications is encrypted using TLS, preventing interception by attackers.
- **End-to-End Encryption:** For sensitive applications, ensure that data is encrypted at rest and in transit, especially when communicating with external services or across untrusted networks.
- **API Security:** Use API gateways and API security tools to secure communications between applications, especially in cloud environments where APIs are frequently exposed.

Enforcing Policies

With the details gathered so far, now you have a clear understanding of the users, devices, and applications in your environment. It is time to think about how to enforce these policies, monitor the users, and continuously refine them. Because there is a complex matrix of users, devices, and applications, it is important to set a baseline and expand from there. This means a base trust with the users and devices is based on certain factors like

- If the user is seen for the first time
- Which factors are used to authenticate and authorize the users
- If there are signs of malicious activities
- If the device is managed; if yes, company-owned or BYOD?

- Device posture details

If these basic checks pass, you can assign the base trust-level policies that allow access to the common and less risky applications. You can then add further levels of trust confirmation for the highly sensitive user and application combination. This means that with base trust users can access common applications such as email and Microsoft Office. For example, if the user login happens at the usual time, using the known devices with an attempt to access the regular applications, the decision engine could assign a high trust level with no additional authorization steps required. This will also help you define trust tolerance, as explained later in this chapter

Granular policies and secondary trust verification are required to access critical assets, including code or financial information. At this point, you also need to decide what actions or changes will result in a loss of trust—for example, a change in hardware. A real-time decision engine is typically used for adaptive access.

Contextual information from the users needs to be collected, stored, and then analyzed to continuously evolve the policies. Advanced AI/ML-based analysis engines could be deployed for this behavior analysis. The information base must have information from events, as described in the following subsections.

Contextual Data

User roles and devices change over time. Employees might change their role for various reasons, such as changing teams or getting a promotion. As a result, the location to access the information also changes. Device type, operating system, and applications used might also change. It is important to collect all this information.

Connection Metadata

It is important to collect user role information and device details at the time of login. User roles could easily be collected from the active directory group memberships, and device details can be collected during posture assessment. Operating system version, installation of mandatory endpoint solutions, device status (e.g., if jailbroken)—all must be collected and stored for further analysis. This also allows you to have a central inventory of all the devices in one place with their security state. This information, when combined with the user information, provides insights into the compliance status of teams and individuals. Necessary actions can be taken if noncompliance is related to a specific group or team.

Logging Suspicion Actions

As you established the baseline trust earlier, it is also important to create the baseline behavior of the users within that trust level. Any deviation from the baseline needs to be recorded and analyzed, and immediate action should be taken. Continuous failed attempts at authorization, use of unsupported platforms to access resources (like jailbroken devices), and attempts to access resources with different user credentials from the same device must all trigger alerts, and immediate action needs to be taken based on company security policies. Additional authorization using MFA, device quarantines, and reduced trust level with each event like this are some of the many actions you can take in these cases. Detailed analysis will help you identify the repeat offenders (intentional or unintentional). AI-based predictive analysis could be used once sufficient data points are available, to help prevent the attacks, reduce the attack surface, and create a self-healing network. Details on self-healing and maturity levels are covered in the later chapters.

Trust Tolerance

Trust tolerance is the degree of flexibility in the zero trust model. While the fundamental principle is “never trust, always verify,” trust tolerance allows for different levels of stringency depending on various factors like the sensitivity of the resource, user context, or current security posture of the device. It answers the question: "How much risk are we willing to accept for this specific access?"

- **High Sensitivity (Low Trust Tolerance):** When a user is accessing critical systems (e.g., financial records, confidential data), there is minimal tolerance for risk. Trust levels must be high, and any small deviation (like accessing from a new location or using a noncompliant device) should trigger additional security measures or block access.

- **Low Sensitivity (Higher Trust Tolerance):** When a user is accessing less critical resources (e.g., internal HR portals or general information), the system can tolerate more variations in trust. For instance, accessing from an unfamiliar device might still be allowed, but with more monitoring.

In zero trust, you will create a dynamic trust tolerance, which adjusts based on the current risk environment. A user trust value could be very high at the time of initial login and authentication. However, as the user tries to access different applications, the trust level can erode based on that user’s behavior. Every organization needs to devise policies that reflect its trust tolerance. Threshold-based dynamic access policies need to be defined. If the trust level of a user is maintained, the sessions and access levels could be extended by a predefined time. In that case of trust falling below a certain threshold, the user needs to be reauthorized before access to the existing or new applications can be granted.

Several tools can help organizations manage trust tolerance within a zero trust architecture:

- Identity and access management (IAM) tools like Okta or Microsoft Azure Active Directory allow for context-based access policies that adjust trust levels dynamically.

- Security information and event management (SIEM) solutions like Cisco Splunk or Elastic Stack help monitor security events and adjust trust tolerance based on real-time risk factors.

- Endpoint detection and response (EDR) solutions like Cisco Secure Endpoint or CrowdStrike provide continuous monitoring of device security, adjusting trust tolerance based on device posture.

By this point, you must have a clear understanding of the overall approach and key decision points to develop a zero trust policy by defining clear objectives around user and device trust, application access, and policy definition requirements. In the next section, we will explore various tools and technologies that help you achieve these outcomes.

Tools and Technologies

When an organization implements a robust zero trust architecture, a variety of tools and technologies are essential for continuously validating users, devices, and applications at every access point. These solutions work together to ensure that only authenticated and authorized entities are allowed to access specific resources. By leveraging these technologies, organizations can enforce strict security policies, limit potential attack surfaces, and create an environment where trust is never implicitly granted but is verified with each interaction.

Central Inventory

A central user and device inventory is like a list that keeps track of who the users are and what corporate and personal devices they are using. Each user and device has a role or job. These roles tell the system what the user or device is allowed to do. This inventory helps the security system decide if a user or device can access certain information or parts of the network. One of the primary ways to implement zero trust is to assign users and devices to specific domains based on their communication needs. Typically, every organization will maintain the active directory for its users, grouped under different business groups. These active directories are then mapped with authentication servers like the Cisco Identity Services Engine, and different policies can then be created for different sets of users. Additional parameters—device type, posture state, time of day, and so on—can be used in authorization policies.

Having a central repository is one of the prime requirements of zero trust implementation. At the time of writing this chapter, Cisco Identity Services Engine can join up to 50 active directory domains. This allows segmented domains to be brought under a single zero trust policy domain.

Identity and Access Management

Validating the user identity using strong methods is fundamental to the zero trust implementation. Passwords can be easily stolen and reused; as such, strong authentication methods such as MFA need to be part of the user validation. Following are some of the approaches you can take:

- **Multifactor Authentication (MFA):** Ensure users provide two or more ways to prove their identity (for example, a password plus a one-time code sent to their phone).
- **cMFA:** Use tools that support continuous validation of the user identity at fixed intervals or based on triggers.
- **Single Sign-On (SSO):** Allow users to log in once and access multiple applications securely.
- **Role-Based Access Control (RBAC):** Set up permissions based on user roles, ensuring people access only what they need. For example, a marketing team member should not access financial systems.

Tools like Okta, Microsoft Azure Active Directory, or Duo Security can help you manage identities and enforce MFA and SSO.

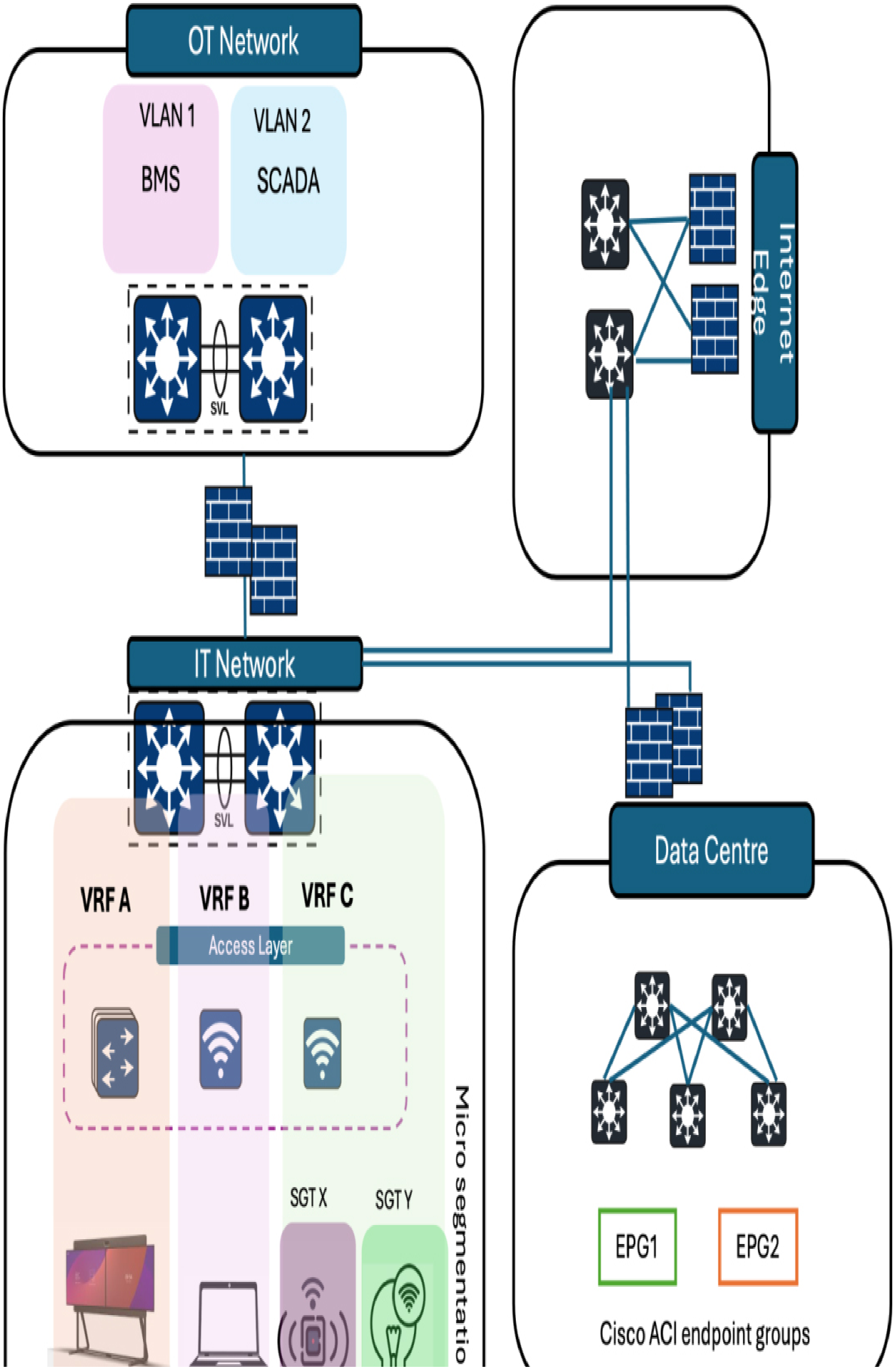
Network Segmentation

To apply the zero trust polices, a network must be segmented using macro- and microsegmentation approaches. Remember that zero trust is not only about providing secure access; it is also about reducing or limiting the impact of any attack. Network segmentation reduces the attack surface by limiting access to a specific domain.

Macrosegmentation is a way to divide a network into smaller virtual networks (segments), usually based on the types of users, devices, or applications. Each virtual network will have its own security rules. For example, you might create one network segment for employees, another for guests, and a third for sensitive data. In zero trust deployment, macrosegmentation helps by limiting who can access certain parts of the network. Even if someone gets access to one segment, that person can’t move freely to other segments without passing additional security checks. This makes it harder for attackers to spread across the network, improving security and reducing the risk of unauthorized access. You will create macrosegmentation using firewall boundaries and virtual segmentation using concepts of virtual routing and forwarding (VRF).

Microsegmentation approaches like VLAN, security group tags (SGTs), or endpoint groups (EPGs) in data centers allow microsegmentation within a macrosegment. The idea is to further restrict and control the communication. With the zero trust principle, only allow communication to what is required. To implement this, smaller network segments are desired. However, the manual assignment of users and devices becomes a management overhead. That is why automated assignment of microsegments is done using AAA servers like ISE.

Figure 3-1 shows high-level macrosegmentation using VRFs and firewalls. You will notice that firewalls are used to isolate the different sections of the network such as the data center, Internet, and BMS/OT areas. The microsegmentation approaches of SGT/VLAN can be used to create microsegmentation within each group. In this example, the enterprise LAN is microsegmented into various segments for corporate laptops, IoT endpoints, and collaboration endpoints. The OT network is using VLAN as a microsegmentation approach with the firewall creating a separate OT zone and macrosegment.


OT Network
VLAN 1
BMS
VLAN 2
SCADA
SVL
Internet Edge
IT Network
SVL
VRF A
VRF B
VRF C
Access Layer
SGT X
SGT Y
Micro segmentatio
Data Centre
EPG1
EPG2
Cisco ACI endpoint groups

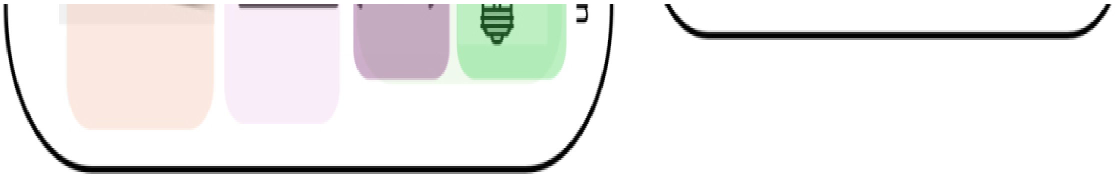


Figure 3-1 *Network Segmentation for a Typical Enterprise with IT, OT, and Data Center Blocks*

Device Posture with Endpoint Security

Because different types of endpoints will connect to different parts of the network, it is important to ensure these endpoints are healthy. Even though the devices will connect in their own microsegments, still they can pose risks to the network and other devices. Accessing the device health and providing that information to AAA servers allow them to be put in the correct microsegments. If a device is detected as unhealthy, usually it is kept in a quarantine zone with access to only remediation tools.

- **Check Device Health:** Ensure all devices that access your network have security updates and antivirus software installed.
- **Device Authentication:** Verify that devices are authorized to access the network. If a device is not secure (e.g., using outdated software), block or limit its access.
- **Endpoint Detection and Response (EDR):** Monitor devices in real time to detect and respond to potential threats.

Tools like mobile device management (MDM), Jamf, or Microsoft Intune can help control device security and access their posture. Cisco Secure Client provides a modular approach with VPN, Posture, and zero trust network access (ZTNA) modules for zero trust deployment.

Virtual Private Network (VPN)

A virtual private network is a technology that creates a secure connection over the Internet. Please note that not all VPN types offer encryption and authentication. In the context of this section, our focus is on SSL/IPsec remote access VPN technologies. VPN allows users to send and receive data as if their devices were directly connected to a private network, even when they're using public networks like Wi-Fi in a coffee shop or hotel. This secure connection is made possible by encrypting the data being transferred and masking the user's IP address, ensuring privacy and security. In a zero trust environment, users and devices must be verified before accessing sensitive resources. VPNs help enforce this by requiring users to authenticate themselves before they can establish a connection. This authentication process adds an extra layer of protection, ensuring that only trusted users can connect. However, you will not be able to apply granular controls with the VPN as demanded by the zero trust. You will note that many companies are moving away from VPN-based access and adopting VPNless secure access using SASE/SSE. VPN is used only for the legacy use cases and that can also be toward the SSE module in the cloud rather than the data center. What it means is that users will connect via VPN into the SSE module with a VPN concentrator sitting in the cloud. Once the user is connected via VPN, it has to go through the regular security service chain before it can access any kind of data either from cloud service providers like SaaS applications or anything in the company data center. This is explained in detail in the section “Applying Zero Trust Using SSE.”

In summary, traditional VPN connections to the organization’s data center allow you to tunnel traffic into the organization, but there is no easy way to apply detailed zero trust checks and policies. Doing so is not impossible, but it does make the design complex. At the time of writing this chapter, the industry is moving toward centralized cloud-based ZTN deployments.

Identifying Business Workflows

At the start of this chapter, we presented an approach to identify and define the trust for users and devices to allow access to different applications using zero trust principles. Now at the time of the deployment, you need to convert them into business workflows. Following are some of the common workflows:

- On-premises employee with a trusted device accessing a private application in the local data center
- On-premises employee with a trusted device accessing a private application in the cloud/SaaS

- On-premises contractor with an untrusted device accessing private applications
- On-premises guests with untrusted access accessing the Internet only
- Remote employees with trusted devices accessing private applications on data center
- Remote employees with trusted devices accessing applications on SaaS

Applying Zero Trust Using SSE

Secure Access Service Edge (SASE) is a cloud-based security model that merges networking and security services into one platform, integrating features like software-defined wide area network (SD-WAN), firewalls, and identity-based access control. Security Service Edge (SSE), a subset of SASE, focuses on security technologies such as Secure Web Gateway (SWG) and Cloud Access Security Broker (CASB). In a zero trust environment, cloud security is crucial as applications move to the cloud. SASE and SSE offer cloud-native tools like CASB to monitor data use and ensure compliance, verifying security continuously even when accessing services remotely. SASE facilitates secure remote connections without traditional VPNs, providing scalable security. SSE safeguards access to cloud apps regardless of user location and, while initially for cloud use, SSE has become the primary method for implementing zero trust due to its simplicity and flexibility. This topic requires detailed discussion for zero trust deployment by any organization in the current era.

Let’s look at the components of SSE before we delve into the details of the deployment strategies. SSE has multiple components such as (but not limited to)

- **Firewall as a Service (FWaaS):** FWaaS delivers firewall functionality via a cloud service, enabling organizations to apply security policies throughout their network, even for remote users. It evaluates and controls traffic flowing between users and applications to thwart threat attacks.

- **Secure Web Gateway (SWG):** SWG safeguards users from web-based threats by filtering out unwanted content, blocking malicious websites, and enforcing acceptable use policies. It also inspects encrypted traffic to ensure that threats aren't concealed within SSL/TLS.
- **Cloud Access Security Broker (CASB):** CASB serves as a security gatekeeper between users and cloud services, implementing security policies, tracking usage, and safeguarding sensitive information in cloud applications.
- **Data Loss Prevention (DLP):** DLP is designed to detect and prevent unauthorized access or sharing of sensitive data. It monitors and controls the movement of sensitive information across the network to ensure it is not leaked or accessed by unauthorized users.
- **Secure DNS:** It helps endpoints from rogue websites. User DNS requests are forwarded to the secure DNS module within the SASE platform, where the URLs are validated against the central database for any vulnerability. Only safe websites are resolved for the users, while URLs hosting malicious content are blocked and the user is notified.

Using SSE, you can deploy zero trust for all common use cases including

- Private application access by the users either on-premises or in the cloud/private cloud
- Secure Internet access
- Legacy VPN-based connectivity to resources both on-premises and in the cloud

As identified in the business workflows, different types of users and device combinations will be used in any organization. Based on the user types, devices can be managed or unmanaged. The SSE architecture typically supports client-based access for managed endpoints and clientless access for unmanaged devices. Cisco Secure Access is an example of SSE that provides secure access to the applications on-premises or in the cloud with zero trust components built in.

Next, let’s look at the different use-case types and deployment approaches.

Client-Based ZTNA Deployment for Managed Corporate Devices

For this first scenario, a zero trust access module needs to be installed on an endpoint for client-based secure access. Cisco Secure clients have a specific module for zero trust. Other vendors have similar functionality, either as a standalone zero trust architecture (ZTA) module or integrated into other endpoint software solutions. The primary function of the ZTA module is to intercept and send traffic to the SSE in the cloud based on policies defined by network administrators. This works as follows:

- The user tries to open any application on their device. The ZTA client sitting on the laptop controls the traffic routing and usually also handles functions like device posture. Typically, you define which applications need to be routed via SSE and which traffic needs to be sent directly to the Internet. These policies are defined in the SSE module and pushed to the ZTA client on the device.

- This traffic is intercepted and sent to the SSE in the cloud. The method to send this traffic to the SSE client is based on the vendor implementation. Cisco uses the QUIC protocol to send traffic to SSE per application. SSE providers usually have multiple points of presence (PoPs) across the planet. Traffic is usually sent to the nearest POP using anycast IP address.

- Traffic first hits the authentication module that decides whether the user/device combination is allowed to access that specific application. Based on the policies defined by the network admin, further authorization flows such as MFA and device posture checks are triggered.

- SSE also has policies that define how traffic needs to be routed toward its destination. The application may reside in the company’s local data center, served from the public cloud, or it can be an SaaS application like Office 365.

- Once the user is authorized, traffic is then routed to the specific destination because SSE usually has direct connections to the specific data (such as IPsec tunnel to the organization's data center, direct high-speed secure connections to cloud service providers).
- The authentication module may trigger periodic reauthorization based on the trust of the device and application access required based on the policies defined in the SSE module.

In this scenario, users can access the required application from anywhere in the world (or from space, as long they have a connection to the SSE portal) without the need for any VPN client. Their application access experience is consistent and without the need to reconnect the company VPN. This method is also known as *VPNless secure access*. Figure 3-2 shows the client-based zero trust access architecture for a corporate device.

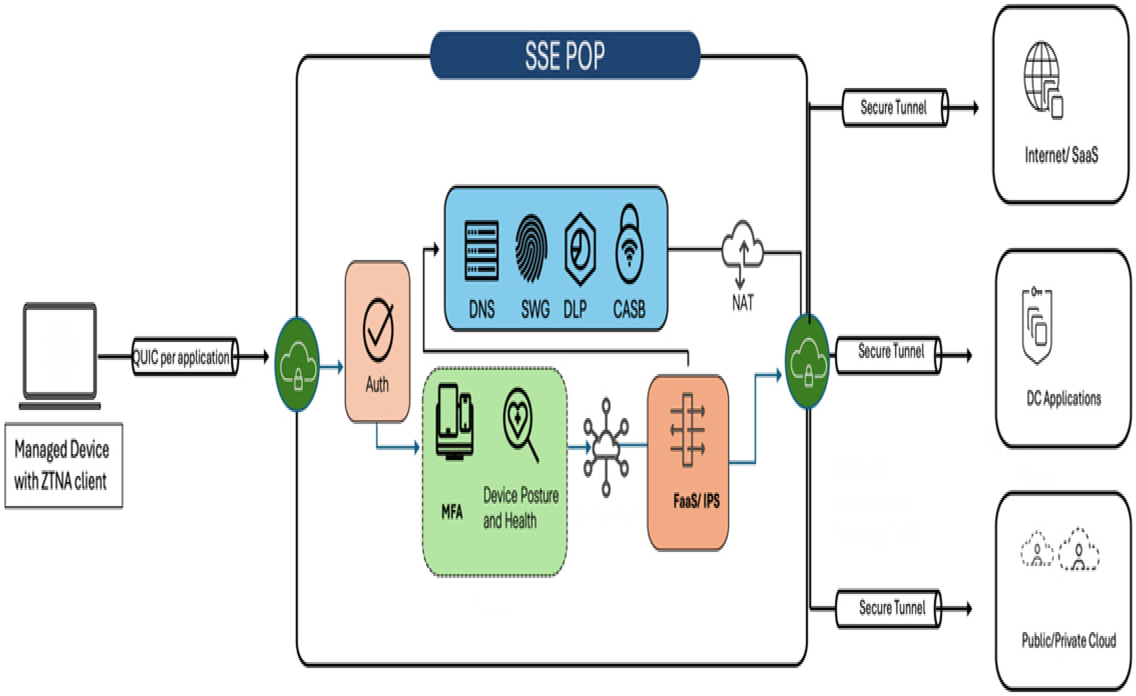


Figure 3-2 *ZTNA Client-Based Access Using SSE*

Clientless ZTNA Deployment for Unmanaged Devices

In the previous example, because the device was managed, it was easy to install the zero trust access module on the client. However, if the user is a partner or guest and the device is not managed, you cannot use the client-based ZTA access methods. In such cases, you need to rely on the browser-based ZTA access. This is also known as *clientless zero trust access*. It works as follows:

1. The user tries to access the application via the browser.
2. Traffic is sent to the SSE via HTTPS tunnels.
3. Based on the authentication and authorization policies, the user is allowed to access a specific set of applications.
4. Traffic is then sent to its destination either in the public cloud, partner data center, or a company data center.

One important difference in this method is that, because there is no ZTA client present on the device posture, information available to the SSE module is limited and based on the browser data only. In such cases, it is recommended to have more restrictive policies. Even in this case, there is no need for VPN clients. Figure 3-3 shows the browser-based access architecture to specific applications in the data center/private cloud. Notice that service chains specific to Internet/SaaS applications have been removed from this flow.

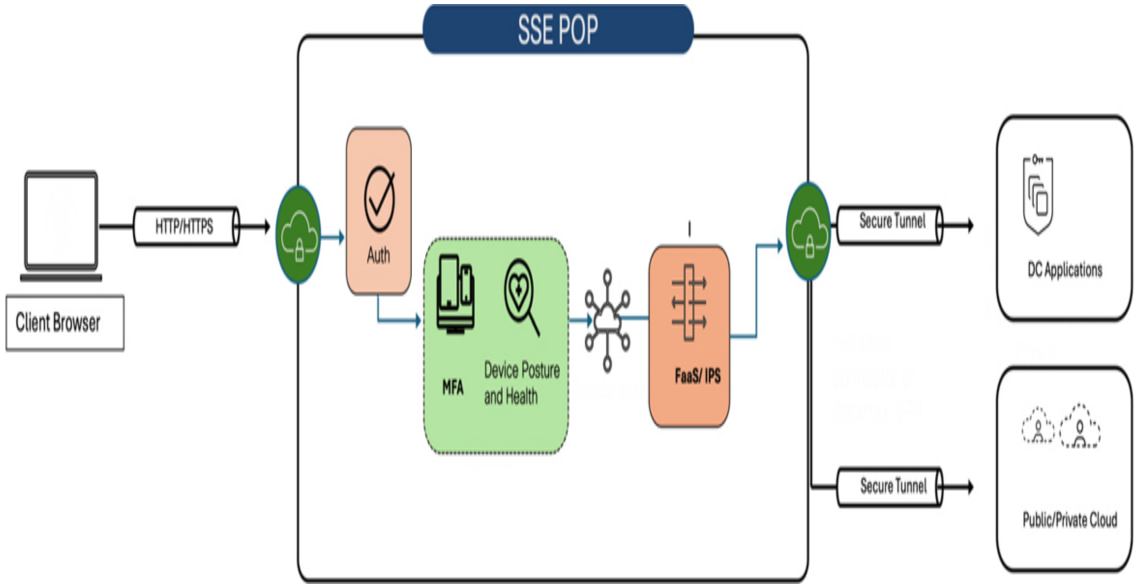


Figure 3-3 *Browser-Based ZTNA*

VPN-Based ZTNA Using SSE

Let's assume that some users require mandatory VPN access. In such cases, you can move your VPN concentrator from the company data center to the SSE cloud. Users will still connect to the SSE via VPN, and from there, security service chains and data policies can remain similar to clientless user access. Figure 3-4 shows the architecture for VPN-based access to private applications only. It is assumed that the Internet/SaaS application could be accessed via split tunneling from the VPN client directly.

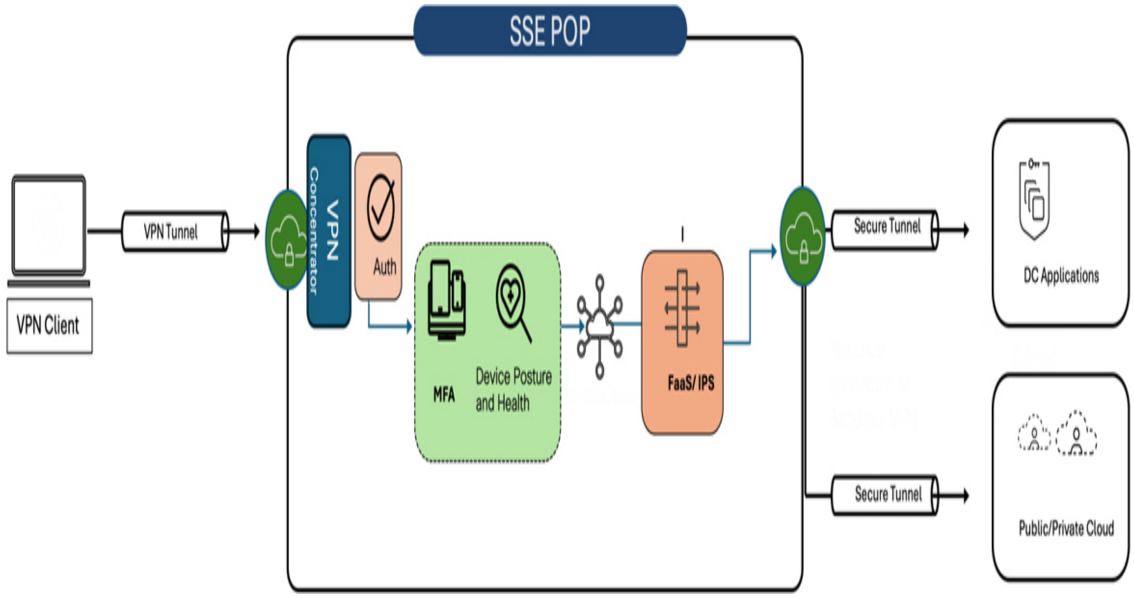


Figure 3-4 *VPN-Based SSE Integration*

SSE Integration for IoT Devices Using SD-WAN

It is not only the users but many devices and IoT devices that need to access their servers via the Internet. You can direct traffic to the SSE in that case also. In such cases, endpoints cannot initiate the tunnels or connections to the SSE POP. Here, technologies like SD-WAN become handy. Assuming that zero trust–based macro- and microsegmentation are already implemented using firewalls, VRFs, and VLANs, traffic from a specific macro- and microsegmentation can then be routed to the SSE using SD-WAN. Most SD-WAN solutions like Cisco SD-WAN can route traffic to a specific destination using a tunneling mechanism based on application type. Cisco SD-WAN calls it *application-aware routing*. You can route traffic toward SaaS applications to SSE and create service chains specific to SaaS or Internet only. Cisco SD-WAN also supports the auto tunneling capability to Cisco SSE.

Once the traffic hits the SSE, POP traffic can then be passed to Internet/SaaS service providers via security service chains or a NAT module as required. Figure 3-5 shows the users and things traffic via SSE for SaaS application access for users and Internet-only access for things like IoT devices.

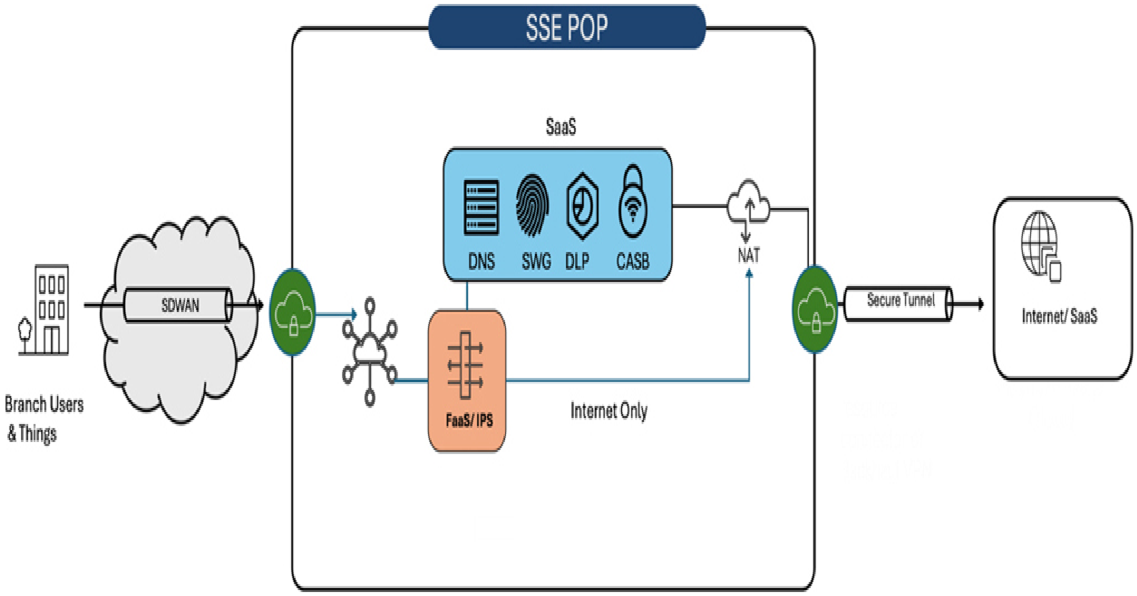


Figure 3-5 *SD-WAN-Based SSE Integration*

ZTNA Deployment Scenarios

Organizations looking to implement zero trust network access (ZTNA) face different challenges depending on whether they have a greenfield or brownfield environment. Greenfield deployments refer to starting from scratch with no existing infrastructure, allowing for a clean, streamlined implementation of ZTNA principles. In contrast, brownfield deployments involve integrating ZTNA into existing, often complex, infrastructures, which requires careful planning to avoid disrupting current operations. Both approaches come with unique considerations, from resource allocation to compatibility with legacy systems, shaping how zero trust principles are applied. In this section, we will look at the high-level strategy for both greenfield and brownfields scenarios.

Greenfield ZTNA Deployment

In a greenfield ZTN, you can build all infrastructure with a zero trust mindset from the start. This allows for a cleaner and simpler implementation, eliminating the need to manage outdated systems.

In the context of deploying zero trust in a greenfield environment, you can take the following steps based on the strategic steps included earlier:

1. Define the zero trust objective:
   - Formulate a distinct vision for the objectives of zero trust within the organization, such as strengthening security, ensuring better compliance, or gaining improved control over the network access.
   - Make sure the zero trust deployment is in sync with overall business goals and strategies. This ensures a strong case for investments and secures executive support.
   - Secure support from senior leadership to guarantee that the initiative receives essential resources and strategic backing import.
2. Define a roadmap:
   - Develop a comprehensive roadmap for implementing zero trust. It must outline milestones, timelines, and essential deliverables. Divide the deployment into manageable phases for organized execution implementation.
   - Establish the budget needed for deployment, factoring in expenses for technology, personnel, and training. Distribute resources appropriately to facilitate each phase of the project deployment.
3. Develop the architecture and design:
   - Develop a high-level architecture that outlines the application of zero trust principles across the organization. This framework should encompass network segmentation, identity management, and access controls.
   - Design the network layout while considering zero trust principles. Ensure segmentation by establishing zones for various types of assets and data.
   - Plan for microsegmentation to limit lateral movement within the network. Define security boundaries for different workloads and services.
4. Deploy:
   - Establish strong IAM systems for overseeing user identities, devices, and applications. Implement multifactor authentication to

enhance access security controls.

   - Create granular access policies based on user roles, device types, and the sensitivity of the resources they are accessing.
   - Implement next-generation firewalls, intrusion detection systems/intrusion prevention systems (IDS/IPS), and secure access gateways.
   - Deploy endpoint protection solutions that include antimalware, encryption, and device management.
   - Ensure that applications are securely developed and deployed. Use application firewalls and secure coding practices.
   - Configure access controls to enforce the principle of least privilege. Use tools to continuously evaluate and enforce access policies.
   - Ensure all data in transit and at rest is encrypted. Use strong encryption protocols and key management practices.
   - Set up comprehensive logging and monitoring to track access, detect anomalies, and respond to security incidents

5. Validate and train:
   - Test access policies to ensure they correctly enforce zero trust principles and do not inadvertently allow unauthorized access.
   - Develop clear documentation and guidelines for users and administrators on zero trust practices and policies.
6. Adapt and update:
   - Regularly update security controls and policies to address emerging threats and changes in the organizational environment.

By following these steps, organizations can effectively deploy zero trust in a greenfield environment, ensuring a secure and adaptive access control framework from the outset.

Brownfield ZTNA Deployment

In a brownfield environment, you deal with established systems, networks, and infrastructure that have been in place for a while. These systems may be outdated and were probably designed without zero trust principles. As a result, there may be existing security vulnerabilities, implicit trust models, or legacy technologies that present challenges for securing them. When you’re implementing zero trust in a brownfield, it's essential to thoroughly assess and adjust the current setup while minimizing disruptions to ongoing operations. The aim is to gradually transition to zero trust, identify weaknesses, and bolster security while ensuring smooth operation. Brownfield settings typically require more phased and adaptable strategies to prevent business interruptions, whereas greenfield environments enable quicker and more seamless zero trust integration since they don't require working around existing infrastructures setups.

You can start with the following steps to define your approach to adopt zero trust in brownfield environment.

1. Define objectives and business needs:
   - Assess the organization’s security goals and what assets are most critical.
   - Conduct a comprehensive assessment of the current security landscape, including existing access controls, user privileges, and device management.
   - Identify where zero trust is most needed; start with high-risk areas such as critical data or sensitive applications.
   - Define the scope of the zero trust implementation, identifying critical applications, sensitive data, and high-risk areas that require immediate attention.
2. Outline the current infrastructure:
   - Conduct a thorough audit of the current network, devices, and user access points.

- Identify gaps where security is lacking or where implicit trust exists that needs to be eliminated.
- Inventory all applications, devices, and users to understand the current access state and any potential issues vulnerabilities.

3. Establish user and device trust:
   - Start by establishing a baseline level of trust for all users and devices, regardless of their current access privileges.
   - Deploy multifactor authentication to strengthen user identity verification.
   - Implement device posture checks to ensure that all devices are secured and compliant before access is granted.
4. Segment the network:
   - Assess whether legacy applications can be segmented or modernized to reduce security risks and enhance overall security posture within the brownfield environment.
   - Apply microsegmentation to limit access to sensitive resources. Each user or device gets access only to the resources they specifically need.
5. Implement adaptive policies:
   - Develop dynamic policies that can adapt based on real-time user behavior, device status, or location.
   - Implement measures to enhance device visibility within the brownfield environment to improve security and reduce risks associated with legacy technologies.
6. Begin with small steps, then scale:
   - Start by applying zero trust principles to a single department or system.
   - Test the framework, gather feedback, and expand gradually across the organization.

7. Continuously monitor and improve:
    - Use security monitoring tools like SIEM to detect anomalies.
    - Regularly review and adjust access controls, policies, and system configurations.
    - Continuously iterate on the zero trust implementation, incorporating feedback, insights, and lessons learned to improve security posture over time

Summary

In this chapter, you learned about the approach for zero trust deployment by first defining the strategy for user and device trust, and application access policies, and then you learned methods for deploying those policies. You also learned about common tools and technologies, including the SSE approach to adopting zero trust. Lastly, you looked at greenfield and brownfield approaches to zero trust deployment.

Chapter 4. Security and Segmentation

In this chapter, you will learn about the following:

- Macrosegmentation and microsegmentation and where to use it
- Security group tag allocation
- Data plane TrustSec deployments
- Control plane TrustSec deployments

Overview

Network segmentation has existed in various forms over the past 30 years. In the early days of the Internet and corporate networks, routed ports were considered plentiful, and switched ports were considered expensive. At this point in time, network separation of resources took place via dedicated, redundant Layer 3 devices (routers) to create service blocks for each distinctive set of applications, devices, and users. This approach allowed operators to deploy rudimentary security in the form of perimeter firewalls or access control lists (ACLs) for each different service block.

With the introduction of multilayer switching for the local area network, the ability to create virtual LANs (VLANs) changed the way that network operators created service separation, resulting in the consolidation of the earlier physical domain separations that were awarded by the (very expensive) approach of building up separate physical blocks on a service basis. While it may seem peculiar today, at that time, the separation of domains with VLANs was considered a level of “segmentation” and potentially even a “security” enhancement in legacy networks.

Such separation that was being performed at the time on the local area network was mirrored on the wide area network through separation of customer-specific traffic flows through frame-relay switching and Asynchronous Transmission Mode (ATM) using data link connection identifiers (DLCIs) and permanent virtual circuits (PVCs) to maintain strict separation between distinct domains or customers, when considering the context of a service provider deployment.

Over the years, further software-based approaches were introduced. These approaches bolstered and added new options to separate not only the broadcast domains of networks but also the respective routing tables and the data plane forwarding paths associated with them through the introduction of tag switching, which later became standardized and vendor agnostic as multiprotocol label switching.

This walk down memory lane—or for many readers, a lesson in the ancient history of computer networking—describes the precursor events that have led to what we today refer to as *macrosegmentation* and *microsegmentation*. This chapter will explore when, where, and how to make the right decisions around which segmentation option is the right approach for your environment.

Segmentation Options

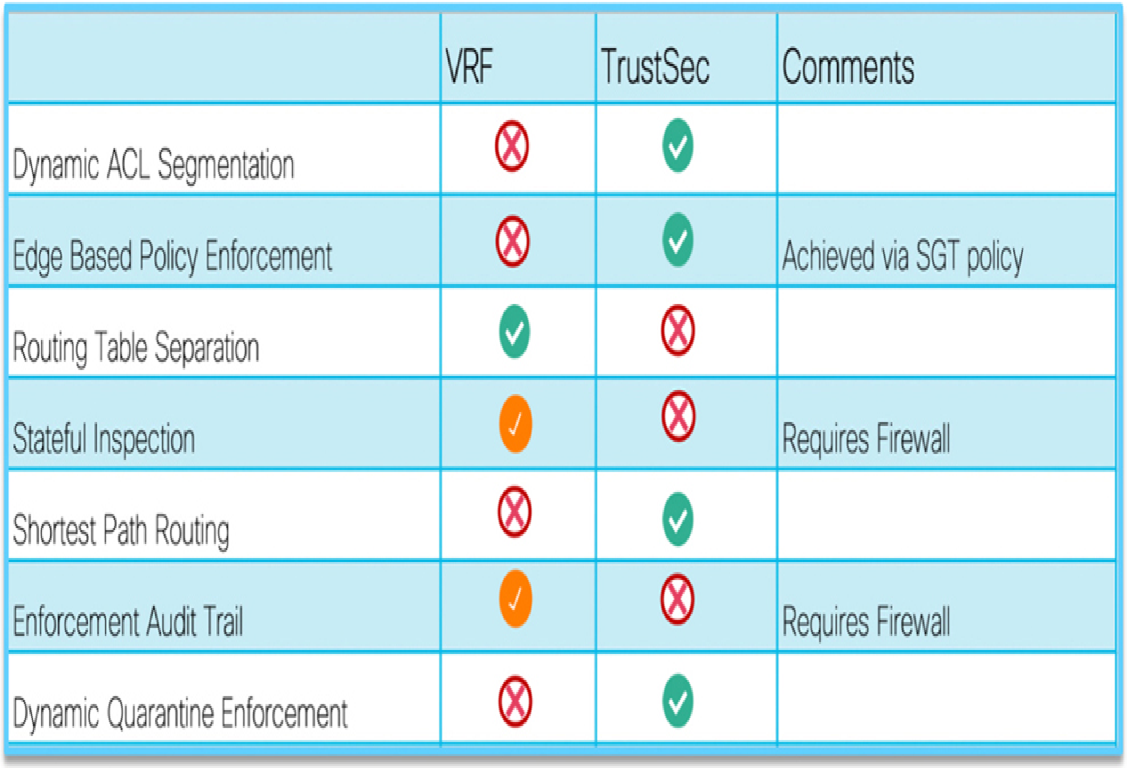
Options that the operator should consider when it comes to segmentation are heavily driven by business requirements, corporate security strategy, and regulatory compliance mandates that dictate which environments they may need to adhere to. Regardless of whether the technology vertical is automotive, healthcare, enterprise, finance, or defense, different factors will need to be weighed against securing the environment versus ensuring operational usability. Finding the right balance is necessary to ensure that the network is both secure and robust, while enabling the operator to manage, troubleshoot, and triage the network.

At times, segmentation policy—and where to segment—is not simply a decision about what is best for the end-to-end architecture and user experience, focusing simply on the tenets and merits of a zero trust design. It also can often result of power struggles within an IT organization or between different siloed teams that are trying to expand, maintain, and control their technology domains or systems.

When an organization is embarking on this journey, it is important to set levels on what the goals and objectives are, and how they transcend department silos, whether they are WAN, LAN, cloud, DC, firewall, or identity teams. Once these goals are aligned between departments, efforts and activities—together with corporate security teams—are often required to ensure that the correct matrix of responsibilities is clear and correctly understood and accepted.

When we are consulting with customers, one question is frequently raised: “Where should I select to use microsegmentation, and where should I consider the use of macrosegmentation?”

The answer to that question often isn’t binary, and there are considerations associated with governance topics and security concepts within the organization. From a general perspective, however, there are a few pros and cons that can be weighed against one another, as summarized in Figure 4-1.



	VRF	TrustSec	Comments
Dynamic ACL Segmentation	⊗	✓	
Edge Based Policy Enforcement	⊗	✓	Achieved via SGT policy
Routing Table Separation	✓	⊗	
Stateful Inspection	✓	⊗	Requires Firewall
Shortest Path Routing	⊗	✓	
Enforcement Audit Trail	✓	⊗	Requires Firewall
Dynamic Quarantine Enforcement	⊗	✓	

Figure 4-1 *Micro- vs. Macrosegmentation Considerations*

Key differences in the two listed approaches are that the use of virtual routing and forwarding (VRF)[nd]based functionalities is directly associated with routed network domains, while security group tags (SGTs) have a correlation to user identity.

Governance Considerations

While macro- and microsegmentation represent segmentation approaches based on software constructs within routers, switches, firewalls, data centers, and cloud hyperscaler environments, the mandate to ensure separation between the domains is sometimes dictated by certain corporate security policies and regulations.

In some very select scenarios where mergers and acquisitions come into play, certain regulators may require that the organizations to be merged remain separate until countries have officially ratified the merger, with government approvals going through.

Macrosegmentation

Following the principles that were defined in technologies such as VRF-Light and MPLS VPNs, the separation of routing tables has provided numerous benefits to customers who are looking to introduce a level of security within their networks.

Historically, many customers went to the lengths of maintaining physically separate hardware to allow for different networks to exist within the same company. This costly and operationally heavy option led to rack space ballooning rapidly as their technical solutions advanced and as their organizations grew.

The use of virtual routing and forwarding instances, as shown in Figure 4-2, sometimes referred to as *virtual networks (VNs)*, resolved many of the problems that these customers faced. Using a function within software, they could create many separate routing tables. This capability opened the door to service providers being able to use duplicate IP addressing and more advanced use cases related to customer traffic engineering.

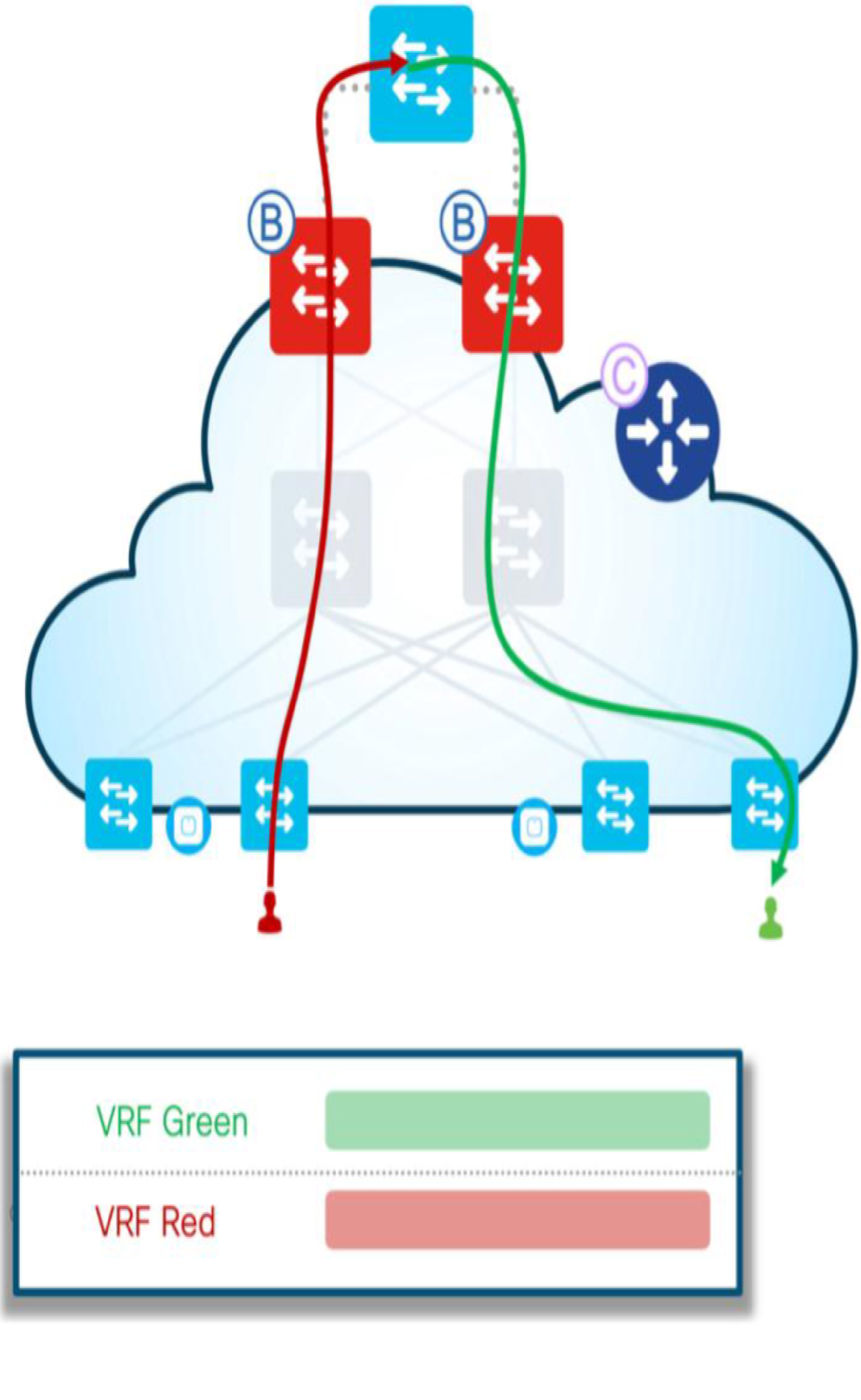

B
B
C
VRF Green
VRF Red

Figure 4-2 *Inter-VRF Traffic Using Macrosegmentation*

With the introduction of macrosegmentation through the use of VRF, customers started to adopt the technology within their networks. Service providers started to replace their legacy ATM and frame-relay switching domains with VRFs through the use of technologies such as MPLS VPNv4 and VPNv6. Enterprises began to adopt the technology as a means to separate key domains of usage within their network.

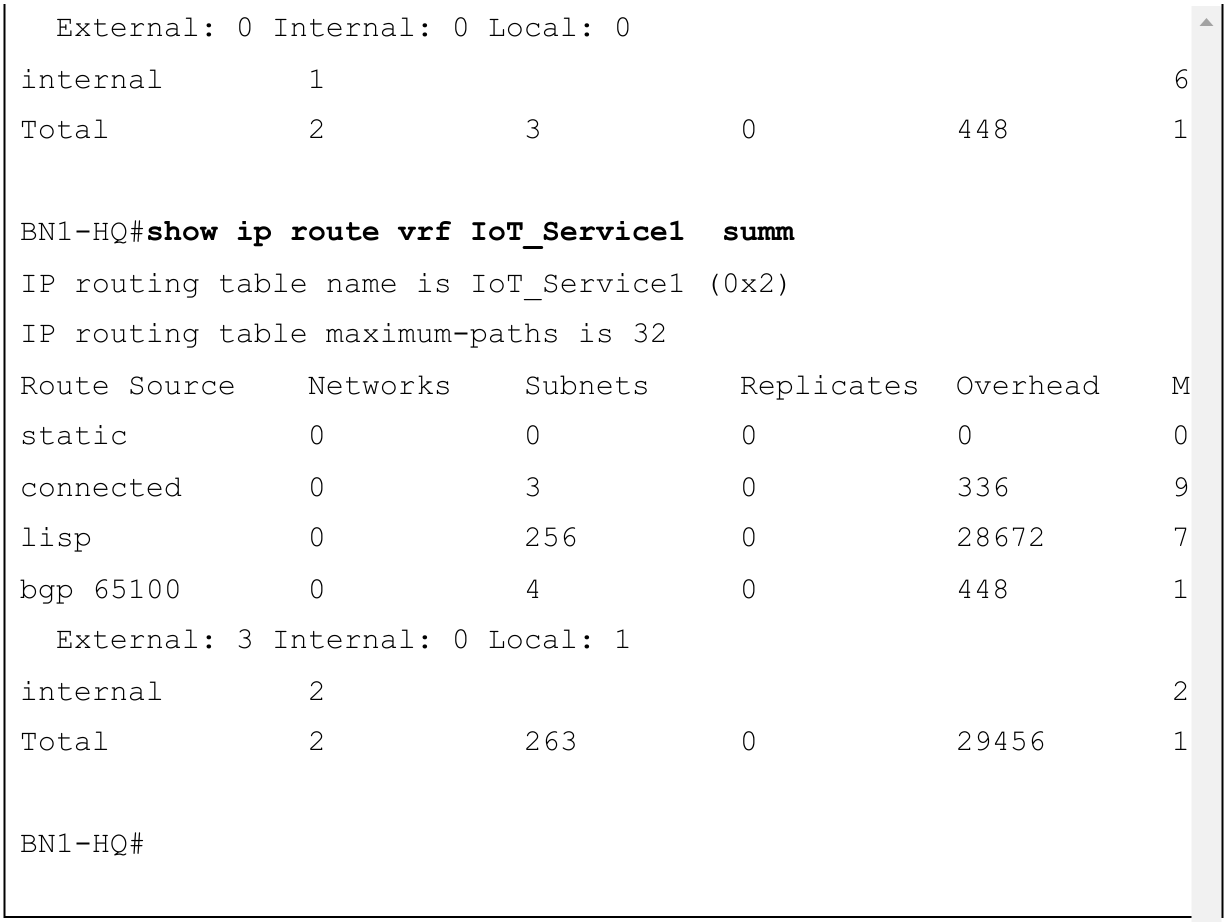
Most commonly, enterprise customers started deploying macrosegmentation for the separation of domains, such as corporate clients from insecure Internet of Things (IoT) devices. In some circumstances, customers also opted to use this approach to separate wired and wireless guest networks into separate domains, ensuring that both corporate and insecure guest devices did not end up within the same routing domain.

At a low scale—two to four virtual networks—enterprise network architectures tend to be easy to manage, and macrosegmentation is a great approach to maintain needed separation. In customer environments that need to scale up to higher numbers of virtual networks, things start to become much more difficult.

In some scenarios, customers with special IoT requirements have attempted to deploy a separate virtual network per IoT service type, as shown in Example 4-1. This approach can become complicated for customers very quickly.

Example 4-1 *Different VN Macrosegmentation Routing Instances on IOS XE*

```
BN1-HQ#show ip route vrf Mgmt-vrf summ
IP routing table name is Mgmt-vrf (0x1)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    M
static          1           1           0           224         6
connected       0           2           0           224         6
lisp            0           0           0           0           0
bgp 65100       0           0           0           0           0
```



```
  External: 0 Internal: 0 Local: 0
internal        1                                               6
Total           2           3           0           448         1

BN1-HQ#show ip route vrf IoT_Service1  summ
IP routing table name is IoT_Service1 (0x2)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Replicates  Overhead    M
static          0           0           0           0           0
connected       0           3           0           336         9
lisp            0           256         0           28672       7
bgp 65100       0           4           0           448         1
  External: 3 Internal: 0 Local: 1
internal        2                                               2
Total           2           263         0           29456       1

BN1-HQ#
```

The challenges that are often voiced by customers in such scenarios are

- Heavy usage of IPv4 addressing due to infrastructure and client subnets
- Routing complexities between domains, often through firewalls
- Manageability challenges for operations teams trying to troubleshoot
- Heavy VLAN consumption and allocation for domain handoffs
- Suboptimal traffic forwarding paths
- Increased size of the failure domain

For service providers, often these challenges are not as common. The reason is not necessarily that macrosegmentation is any different; rather, it is often deployed simply to create an end-to-end network per customer. The nuances and challenges associated with leaking and routing between VRFs are often not factors that they need to consider in many cases.

To further simplify deployments, service providers often use grade standardization and automation via tools such as Network Services Orchestrator (NSO). This links in with customer provisioning portals and business process automation systems, making the provisioning and deprovisioning of such networks almost fully autonomous for the operators within the service provider. Across domains, in modern architectures, there is also an increasing demand to shift to an Infrastructure as Code (IaC) approach for performing network provisioning, deployment, and testing. This approach is often achieved through the use of tools such as Ansible and Terraform.

Routing Paths

When looking at macrosegmentation, network teams often use the terms *tromboning* of traffic or *hairpinning* of traffic flows negatively when making use of VRFs. In more simplistic terms, this actually means that the source device, whether a laptop or an IoT device, must send traffic a long distance through the IP network to once again return to a physically adjacent or close location, sometimes even the same switch. While it is indeed possible that in certain network architectures and topologies traffic may take such suboptimal routing paths to get from one node to another, it certainly is not always the case.

In a well-planned network, the resources that should communicate with one another should normally be structured in a manner that allows for a relatively short path to be taken. The purpose is to avoid unnecessary network bandwidth utilization and lower the potential failure domain, in the case of an outage scenario within the path.

Figure 4-3 illustrates how a network with hairpinned traffic could look in terms of its forwarding path.

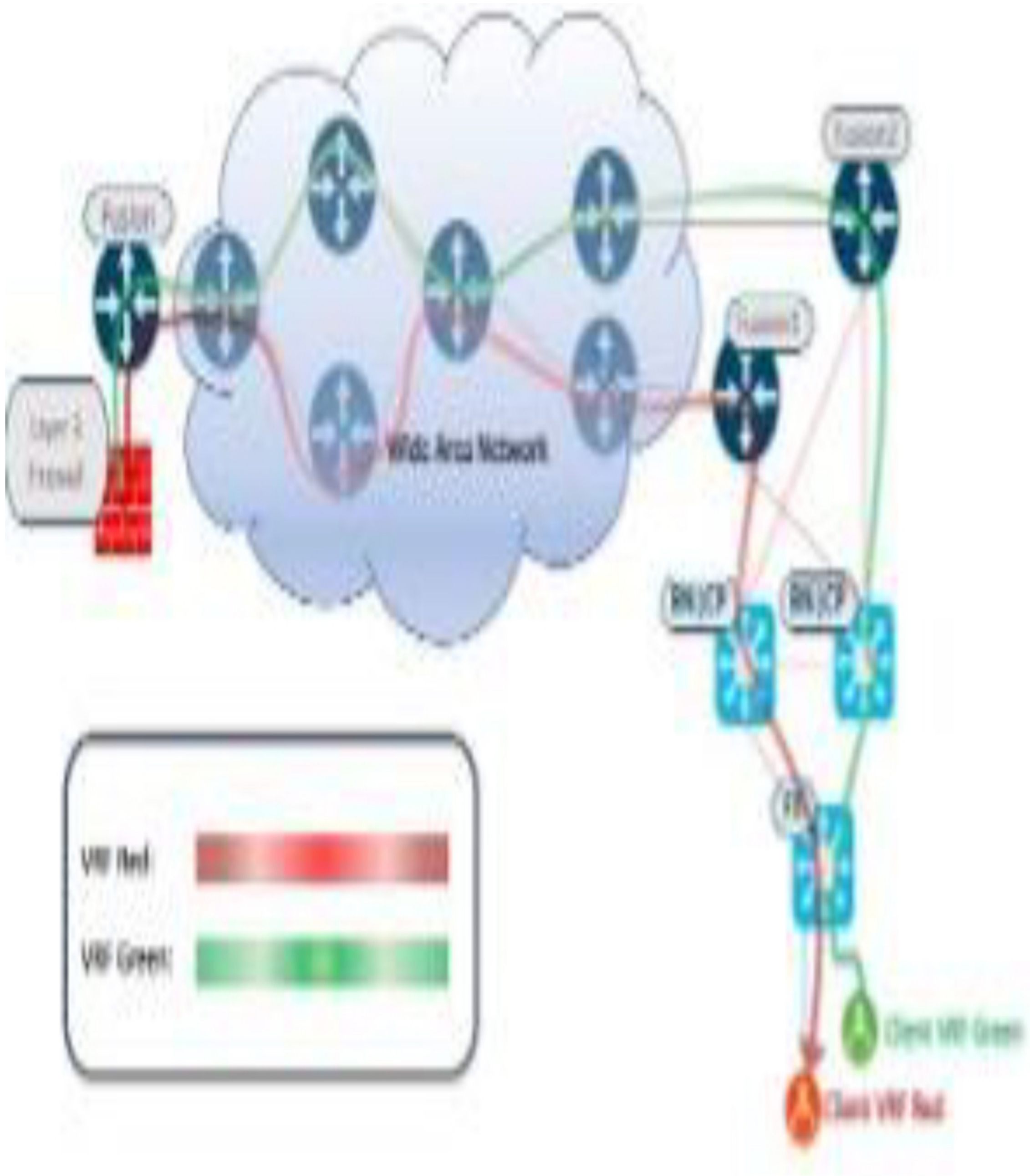


Figure 4-3 *Suboptimal Routing*

Stateful Inspection

While virtual networks are able to separate traffic using dedicated routing domains, they do not provide an ability to actually look inside the traffic, or permit and block flows of traffic between systems at an application level. This is where stateful inspection comes in.

Stateful inspection enables the network operator and security teams to inspect within an application traffic flow to identify patterns that may match malicious behavior and to enforce network actions on the matched traffic based on a specific security ruleset that is configured.

Often, between different security domains within customer organizations, there are mandates to perform stateful inspection. In particular, this request is seen for customer networks between IOT devices and their controllers, between IT and OT networks, or between data center resources and the clients that are connecting to them.

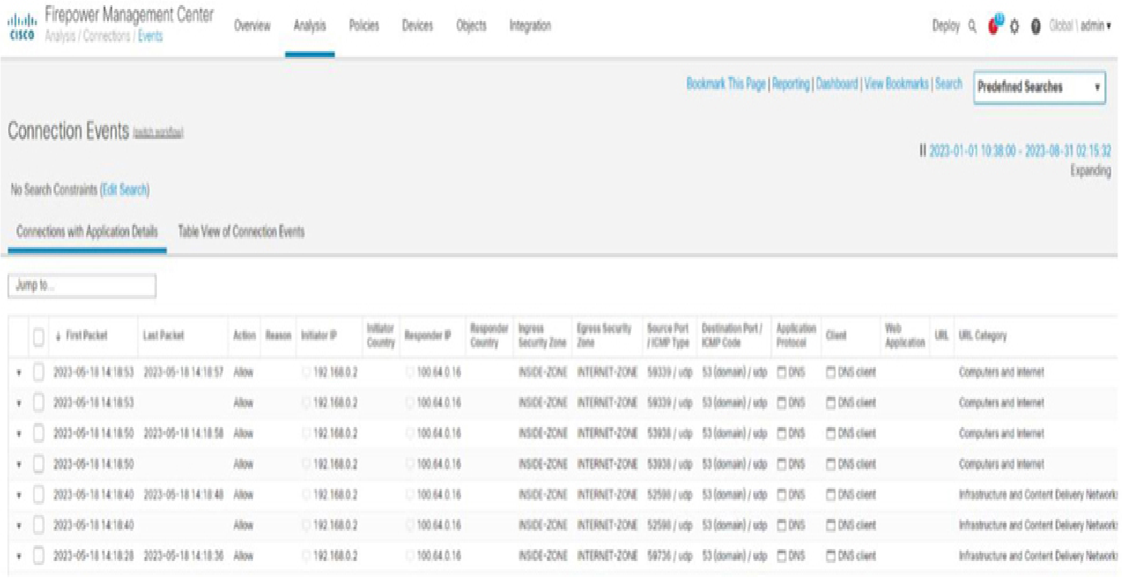
So how does this relate to macrosegmentation exactly? In short, it doesn’t. However, if we revisit the earlier topic describing the hairpinning and tromboning that happens between networks using VRFs, this capability can help in the context of steering traffic between domains through a firewall for stateful inspection.

The fact that VRFs maintain a strict separation of routed traffic means that if traffic needs to go between networks, it needs to merge at a central point. Merging two VRFs at a central point without a security appliance or device essentially removes all the segmentation benefits that were initially provided by creating the separation in the first place. For this reason, best practices for communications between domains require the use of a form of security.

Audit Trail

In many deployments, it is critical to be able to track down a historical view of which traffic, network address translations, and security group tags and even user identity mapping traversed a firewall; this critical piece of information needs to be captured. Many reasons exist for why this information needs to be captured: from regulatory requirements for retention of specific data in financial services domains to the need for correlation with other systems from an operational perspective to provide the right insights in outage situations. There are even scenarios where police or intelligence agencies may require the ability to request the data for traffic that may have traversed a corporate network at a particular date and time to initiate legal proceedings against an individual when illegal activities may have taken place.

To have such a trail of data, the use of a firewall between domains, as shown in Figure 4-4, can facilitate such audit events being captured.


Firepower Management Center
Analysis / Connections / Events
Overview
Analysis
Policies
Devices
Objects
Integration
Deploy
Global \ admin
Bookmark This Page | Reporting | Dashboard | View Bookmarks | Search
Predefined Searches
Connection Events
No Search Constraints (Edit Search)
Connections with Application Details
Table View of Connection Events
Jump to...

II 2023-01-01 10:38:00 - 2023-08-31 02:15:32
Expanding

First Packet	Last Packet	Action	Reason	Initiator IP	Initiator Country	Responder IP	Responder Country	Ingress Security Zone	Egress Security Zone	Source Port / ICMP Type	Destination Port / ICMP Code	Application Protocol	Client	Web Application	URL	URL Category
2023-05-18 14:18:53	2023-05-18 14:18:57	Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	59339 / udp	53 (domain) / udp	DNS	DNS client			Computers and Internet
2023-05-18 14:18:53		Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	59339 / udp	53 (domain) / udp	DNS	DNS client			Computers and Internet
2023-05-18 14:18:50	2023-05-18 14:18:58	Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	53938 / udp	53 (domain) / udp	DNS	DNS client			Computers and Internet
2023-05-18 14:18:50		Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	53938 / udp	53 (domain) / udp	DNS	DNS client			Computers and Internet
2023-05-18 14:18:40	2023-05-18 14:18:48	Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	52598 / udp	53 (domain) / udp	DNS	DNS client			Infrastructure and Content Delivery Networks
2023-05-18 14:18:40		Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	52598 / udp	53 (domain) / udp	DNS	DNS client			Infrastructure and Content Delivery Networks
2023-05-18 14:18:28	2023-05-18 14:18:36	Allow		192.168.0.2		100.64.0.16		INSIDE-ZONE	INTERNET-ZONE	59736 / udp	53 (domain) / udp	DNS	DNS client			Infrastructure and Content Delivery Networks

Figure 4-4 *Audit Trail of Events in a Firewall*

Note

It is possible to collect such communication flows from other systems in the network to also provide a viable audit. Systems such as Secure Cloud Analytics can also provide useful data to meet such needs and requirements.

Failure Domain

Most likely, a logical association with the deployment of a larger number of hops occurred between a source and destination system for many readers. The more systems within an end-to-end path, the higher the probability of one of those intermediate systems failing. This doesn’t necessarily mean that having a longer end-to-end communication path is always negative, but rather, when the potential size of the failure domain is increased to a broader reach, it remains critical that the right levels of resilience and testing are introduced and validated on a regular basis for their efficacy.

Microsegmentation

In the earlier section, we highlighted many of the benefits and caveats associated with the use of macrosegmentation. While the technology itself provides many gains, an area where it struggles to deliver the needed capabilities at scale is in maintaining the separation of many different devices or identities without having to send traffic through a central point(s) in the network to converge. This challenge is the area in which microsegmentation excels.

Unlike macrosegmentation, in microsegmentation, the separation between domains is not achieved based on an IP-routed boundary, an IP subnet, VRF, or the need to traverse a central firewall or route-leaking point. Rather, the key attribute that is relevant in the context of microsegmentation is the mapping of an identity to a tag; in real-world terms, this would be comparable to an individual's driver's license or passport, which confirms their identity to authorities when they travel. In Cisco products, depending on the associated platform, this tag can be referred to as a security group tag (SGT) or endpoint group (EPG) when being applied in the context of ACI data center deployment.

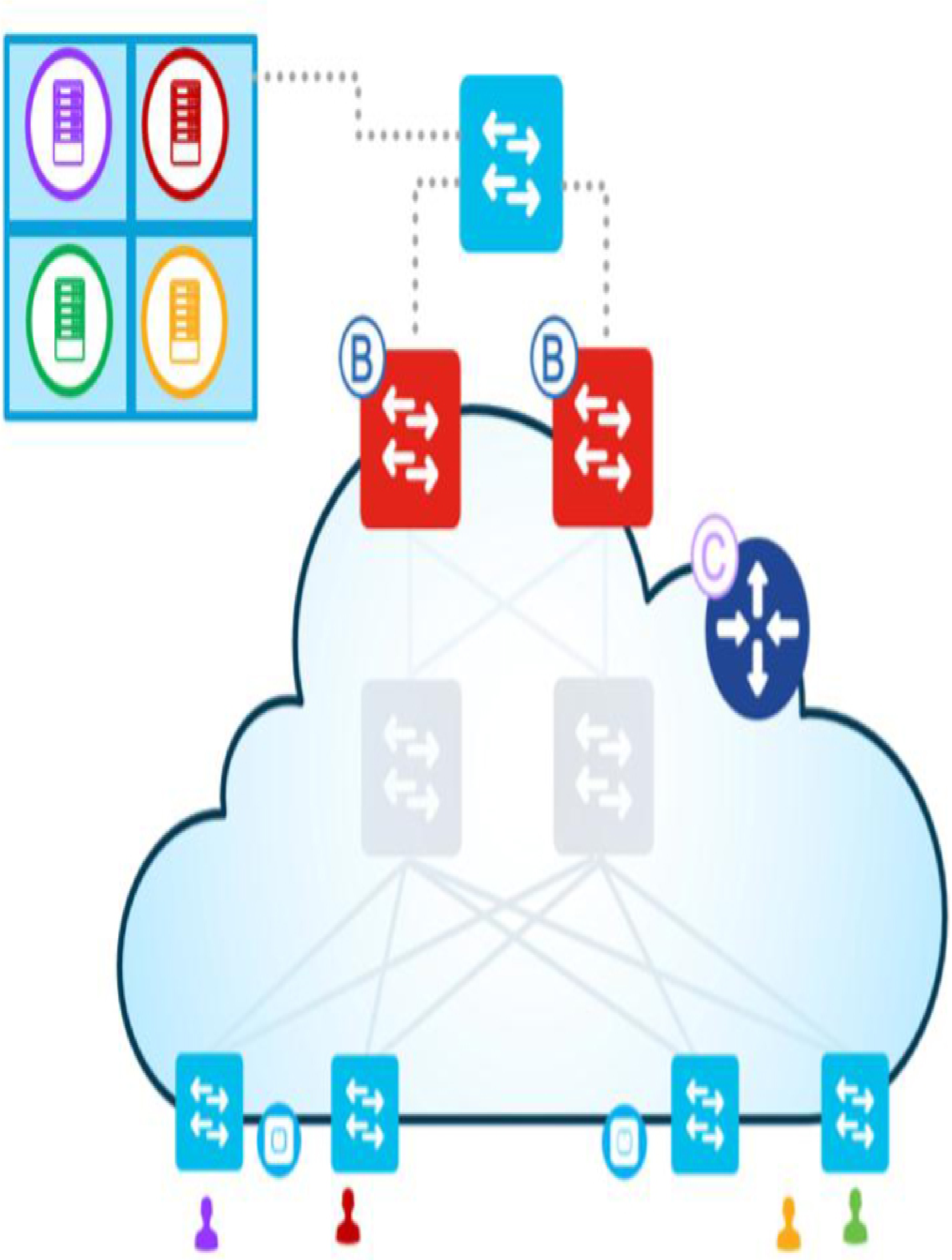
The allocation of a tag or group based on the identity of a system or user to each and every device within a network provides a much more granular ability to be able to perform segmentation. The deployment and enforcement of security group tag architectures within Cisco products are collectively referred to as Cisco TrustSec. TrustSec not only allows the operator to allocate a tag based on identity attributes to systems but also allows for the enforcement of a security policy restricting or permitting communications between different systems.

> **Note**
>
> In earlier deployments of Cisco TrustSec, the naming convention *source group tag* was used; this name has since been deprecated.

You can see an example of how communications may look within a Cisco TrustSec network in Figure 4-5. The differently shaded user icons in different colors represent individuals that map to a particular group, within a single IP subnet. In a macrosegmentation domain, without specific access lists in place to restrict communications, traffic would be sent automatically between systems.

When TrustSec is used with security group tags allocated to the clients in this network, the peer-to-peer communications can be blocked, permitted, or selectively permitted based on the security team's or network administrator's preferences. Let's look at the example of the light grey host; it has an SGT of 100 with an SGT name of K8SDevEnv1. This particular tag is used for developers who should only be permitted access to the Kubernetes development cluster, which is connected with the dark grey icon at the top left of Figure 4-5. With the correct rule in place, this client device can communicate with its respective development cluster without access to any other systems.

A similar configuration could be applied to the other workstations, which have been allocated SGTs for the IOT-ACS and CorporatePCs tags.



Aggregate Range 10.0.0.0/16

Agency A	Name: K8SDevEnv1	SGT 100
Agency B	Name: K8SDevEnv2	SGT 200
Agency C	Name: IoT-ACS.	SGT 300

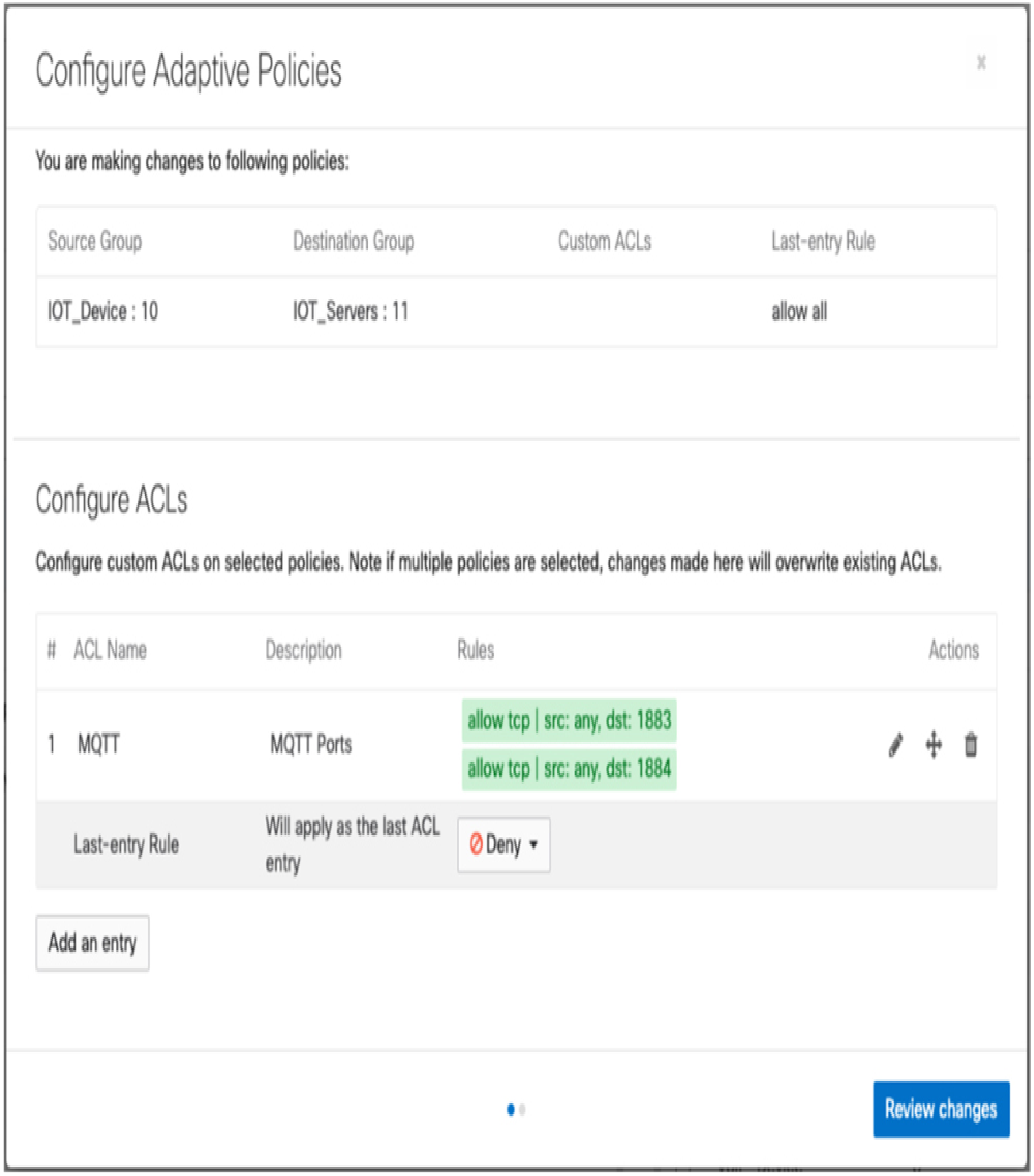
Agency D
Name: CorporatePCs
SGT 400

Figure 4-5 *Security Group Tag Deployment*

The advantage to this approach is that for each and every client device, IOT device, or even server, an SGT mapping could be applied to align to the relevant zero trust principles of the organization. This approach limits the ability for lateral movement between systems to take place, mitigating the potential for certain exploits to take place from would-be attackers in many circumstances.

Mapping the identity of the endpoint or the user to a specific system also fosters the ability to raise or reduce the user’s privileges to the network based on their role, department, tenure, project engagement status, or deployed system state. All of this is based on a policy matrix that exists on the respective platform, providing a dynamic lower level of privilege when not being updated with the latest security patches.

Configuration policies and contracts can be applied in a similar method to legacy IPv6 and IPv6-based access lists on Cisco products. However, rather than selecting a source and destination IP address or address range, you can define the source group and destination group tag as depicted in Figure 4-6.



Configure Adaptive Policies

You are making changes to following policies:

Source Group	Destination Group	Custom ACLs	Last-entry Rule
IOT_Device : 10	IOT_Servers : 11		allow all

Configure ACLs

Configure custom ACLs on selected policies. Note if multiple policies are selected, changes made here will overwrite existing ACLs.

<table>
<tr><th>#</th><th>ACL Name</th><th>Description</th><th>Rules</th><th>Actions</th></tr>
<tr><td>1</td><td>MQTT</td><td>MQTT Ports</td><td>allow tcp | src: any, dst: 1883
allow tcp | src: any, dst: 1884</td><td></td></tr>
<tr><td></td><td>Last-entry Rule</td><td>Will apply as the last ACL entry</td><td>Deny ▾</td><td></td></tr>
</table>

Figure 4-6 *Adaptive Policy Configuration for Cisco Meraki Deployments*

Policies associated with Cisco TrustSec can be configured in a variety of places and are often depicted in a matrix format, as shown in Figure 4-7 and Figure 4-8. This approach simplifies the view of the security teams and/or network operators. Configuration is also possible on the command-line interface (CLI) of many router, switch, and firewall platforms. Where the configuration for the deployment of TrustSec tags and policies takes place can depend on user preference, and architectural decisions that may be made in the future for a deployment.

The use of Cisco Catalyst Center or the cloud-based Cisco Meraki solution, where there is no on-premises orchestrator required for command and control activities, but rather a cloud-based tether to network components that are being managed, monitored, and orchestrated.

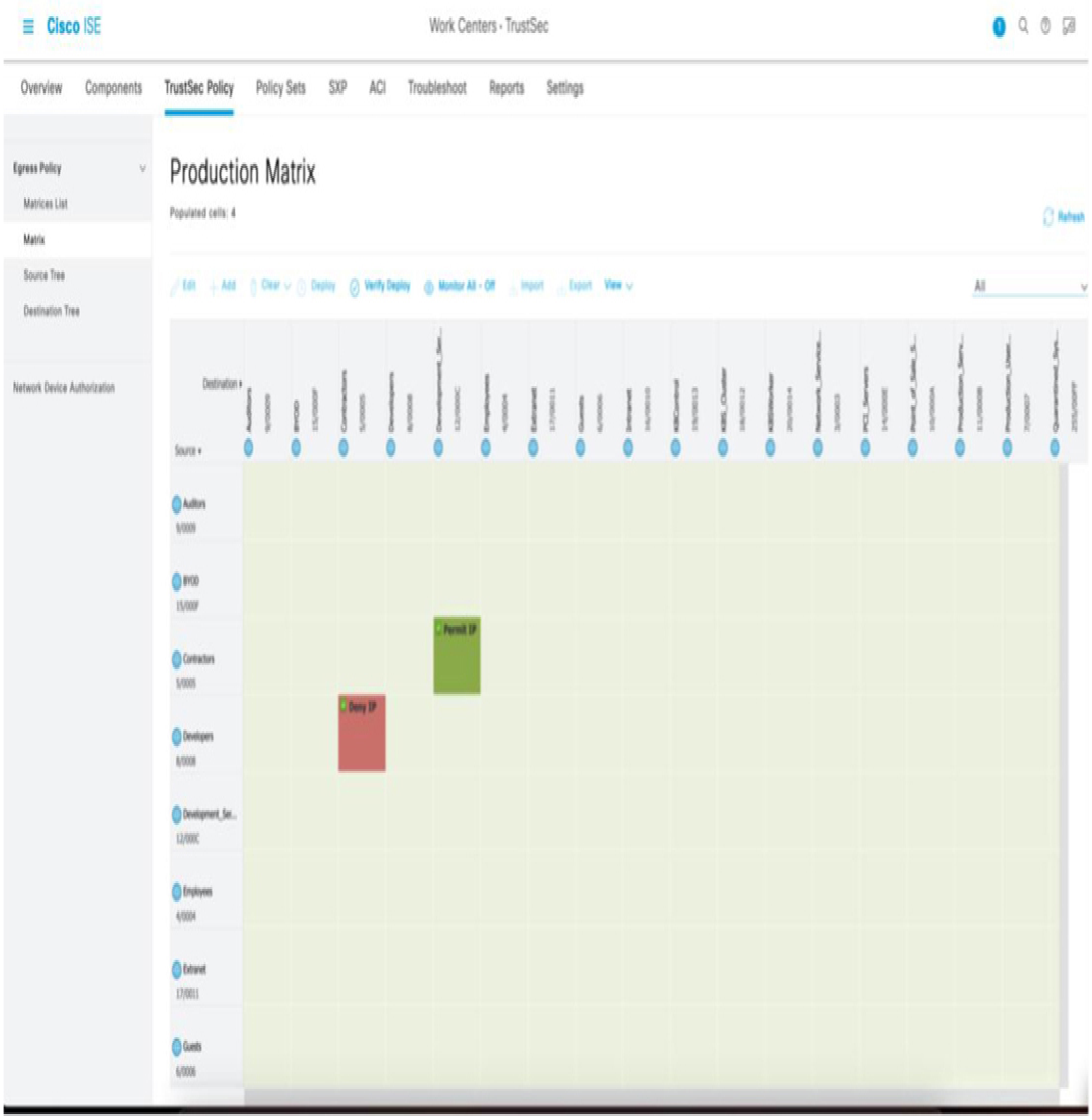
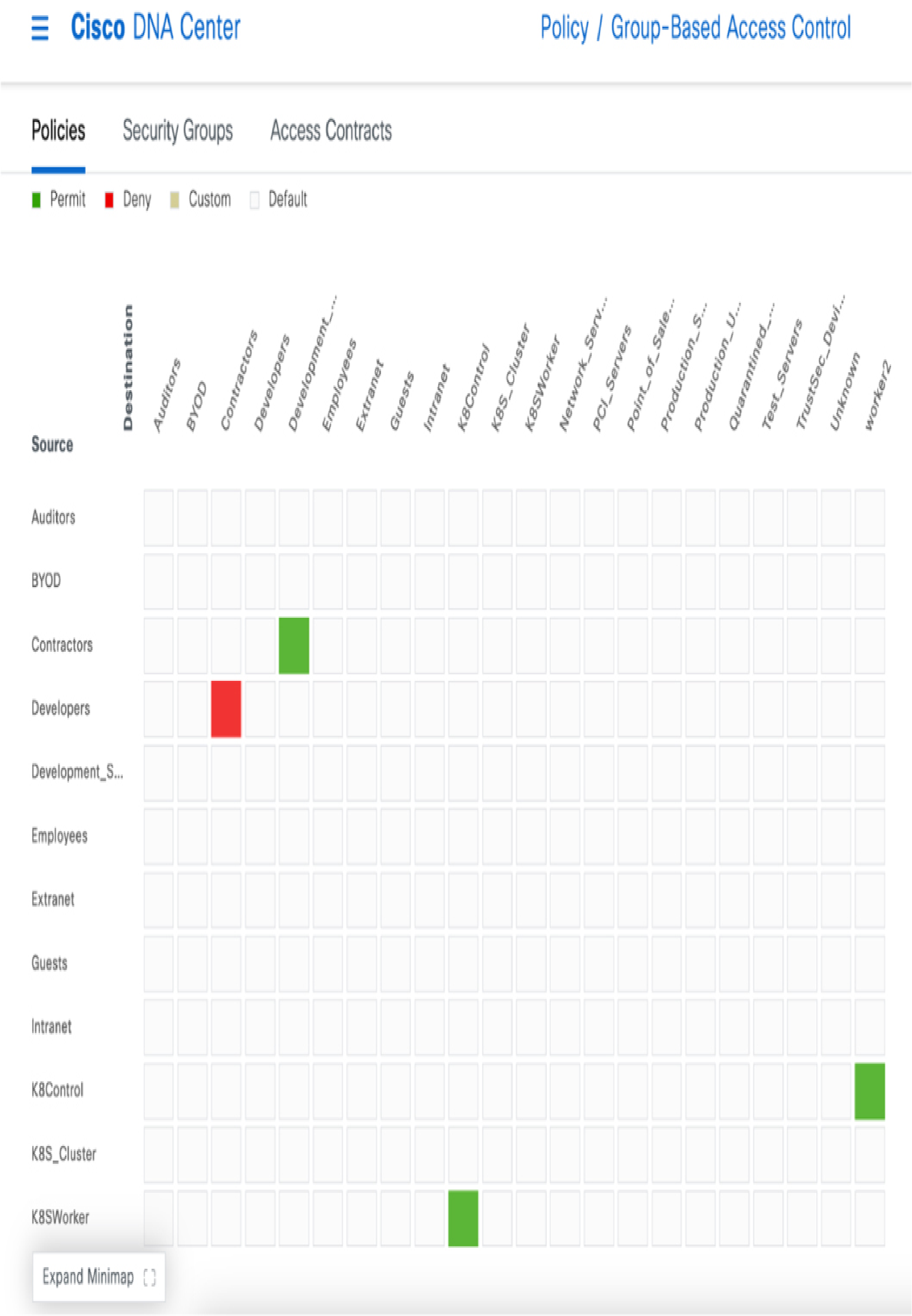


Figure 4-7 *TrustSec Matrix on Identity Services Engine*

Note

Certain features such as Inter-cluster SDA Transit and Inter-cluster LISP Extranet may mandate that Cisco Catalyst Center be selected for the TrustSec policy configuration instead of the Cisco Identity Services Engine.



🟩 Permit 🟥 Deny ▨ Custom □ Default

Destination Source	Auditors	BYOD	Contractors	Developers	Development_...	Employees	Extranet	Guests	Intranet	K8Control	K8S_Cluster	K8SWorker	Network_Serv...	PCI_Servers	Point_of_Sale...	Production_S...	Production_U...	Quarantined_...	Test_Servers	TrustSec_Devi...	Unknown	worker2
Auditors	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
BYOD	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Contractors	□	□	□	□	🟩	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Developers	□	□	🟥	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Development_S...	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Employees	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Extranet	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Guests	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
Intranet	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
K8Control	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	🟩
K8S_Cluster	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
K8SWorker	□	□	□	□	□	□	□	□	□	🟩	□	□	□	□	□	□	□	□	□	□	□	□

Figure 4-8 *Group-Based Policy Matrix in Cisco Catalyst Center*

Best Practices for Macro- and Microsegmentation

When transitioning from a classic network architecture, which historically in campus networks have often run a flat global routing table using OSPF as a routing protocol and relying on Spanning Tree for Layer 2 loop prevention, an organization needs to take a number of considerations into account. The same applies for networks shifting from a classic approach of deploying IPv4 and IPv6 interface ACLs for enforcement of data plane communications toward the use of microsegmentation.

Key considerations for macrosegmentation should be split into two different areas. First, what needs to be considered during the migration phase, which, depending on the size of the network, could take over a year to migrate to the new mode of operation? Second, post-migration, is there a need for inter-VN communications? If yes, does the communication from one VN to another represent a requirement to increase the bandwidth in certain parts of the network due to potential suboptimal forwarding patterns, and will the new forwarding patterns potentially increase the latency negatively in time-sensitive applications?

Following are some important considerations when migrating to macrosegmentation:

- Until the complete network is migrated, which points in the network will represent the leaking between routing tables?
- To avoid routing loops in the network, are the correct routing policies in place to avoid feedback from different VNs?
- How will the temporary traffic patterns look while migrating the network to use VRFs? Do redundant paths still exist? Has the resilience concept been tested?
- Do dependent security components need to be adjusted in any way? Firewalls, load balancers, and so on?
- Which subnets or services need to be leaked from the global routing table to the virtual networks?

- Which methods will be applied (ACLs/firewall) to ensure security is maintained during the transition phase and to ensure that clients in the global routing table cannot access VNs that they should not have rights to communicate with?

Once migrated to macrosegmentation, these further topics need ongoing consideration:

- What is the policy for the generation of a new VRF/VN?
- How should address planning take place to avoid suboptimal routing?
- At which point should there be a reevaluation of placement for workloads, services, and systems when increased bandwidth demands are observed?
- Which traffic types need to traverse between VN/VRFs?
- How should the redundancy concept look like for inter-VN traffic? Regional, global, both?

For microsegmentation, there are also a number of considerations that need to be taken into account when migrating and operating the architecture. During the migration phase, the following should be considered:

- How will security enforcement take place until the network is fully migrated (considering that security is enforced on the egress)?
- What is the right balance in terms of policy allocation that maps to the organizational needs and operational maintainability?
- Should the migration phase only perform a monitoring function, or is an enforcement function preferred?
- What is the corporate policy for the creation and allocation of a new microsegmentation domain (or SGT)?
- Is inter-VRF microsegmentation desired, or should enforcement between VRFs take place on a security appliance?
- Will a default permit or default deny model be selected, and does the hardware platform selected allow for the future scale needs in either

mode?

- Will external services such as cloud and data center also have microsegmentation allocation and enforcement?
- Is there a definition of golden templates for microsegmentation configurations that may exist outside the domain of an orchestrator for interfaces using inline tagging or systems provisioned with SGT Exchange Protocol (SXP), and are there regular checks to ensure that these configurations are consistent?

Upon conclusion of the migration phase, the ongoing operation of the network using microsegmentation needs to be considered:

- How are policy hits monitored within the architecture? Do operator actions take place in the case of an unusual peak of permits or denies for a given policy?
- How often are microsegmentation tags and policy reevaluated for decommission or enrichment?
- Are flows being enforced on a security appliance like a firewall that should transition to microsegmentation enforcement or vice versa?
- Is there a need to monitor the ongoing scale of platforms to ensure that capacity limits are not breached for policy download?

The preceding list of practices is not necessarily a one-size-fits-all topic, and decisions around which actions to pursue can vary from organization to organization. That being said, a network tends not to be a static entity but rather an evolving mass transit system for the communications of business relevant and critical services. Because these services and their traffic patterns change and morph over time, so do the requirements to evaluate and reevaluate the policies, rulesets, tags, and segmentation functions that are applied for ongoing veracity.

Verification of Security Group Tags on IOS XE Platforms

The mappings of SGT names to tags within Cisco IOS XE platforms can be verified on the CLI when you are logged in to the system via Telnet or SSH. You can use the CLI command **show cts environment-data** to verify the deployment of Cisco TrustSec Configurations and the corresponding security group tag names that have been retrieved from the AAA Radius Server, as shown in Example 4-2.

Example 4-2 *SGT Overview on Cisco IOS XE*

```
Edge#show cts environment-data
CTS Environment Data
====================
Current state = COMPLETE
Last status = Successful
Service Info Table:
Local Device SGT:
  SGT tag = 2-00:TrustSec_Devices
Server List Info:
Installed list: CTSServerList1-0001, 1 server(s):
 *Server: 192.168.70.101, port 1812, A-ID 2781F092AD350599EE43C12
          Status = ALIVE
          auto-test = TRUE, keywrap-enable = FALSE, idle-time = 6
Security Group Name Table:
    0-00:Unknown
    2-00:TrustSec_Devices
    3-00:Network_Services
    4-00:Employees
    5-00:Contractors
    6-00:Guests
    7-00:Production_Users
    8-00:Developers
    9-00:Auditors
    10-00:Point_of_Sale_Systems
    11-00:Production_Servers
    12-00:Development_Servers
    13-00:Test_Servers
    14-00:PCI_Servers
    15-00:BYOD
    16-00:Intranet
    17-00:Extranet
    18-00:K8S_Control
    19-00:Worker1
    20-00:Worker2
    21-00:Worker3
    22-00:K8_DEVS
    255-00:Quarantined_Systems
Environment Data Lifetime = 86400 secs
Last update time = 07:00:28 UTC Thu Aug 31 2023
Env-data expires in   0:18:31:41 (dd:hr:mm:sec)
Env-data refreshes in 0:18:31:41 (dd:hr:mm:sec)
Cache data applied           = NONE
State Machine is running
Retry_timer (60 secs) is not running
```

To see the per IPv4 and IPv6 to SGT Mapping overview, you can use the **show cts role-based sgt-map vrf <vrf name> all** command, which provides an overview of the SGTs known to the platform and the source of origination for the IPv4 or IPv6 prefix (see Example 4-3).

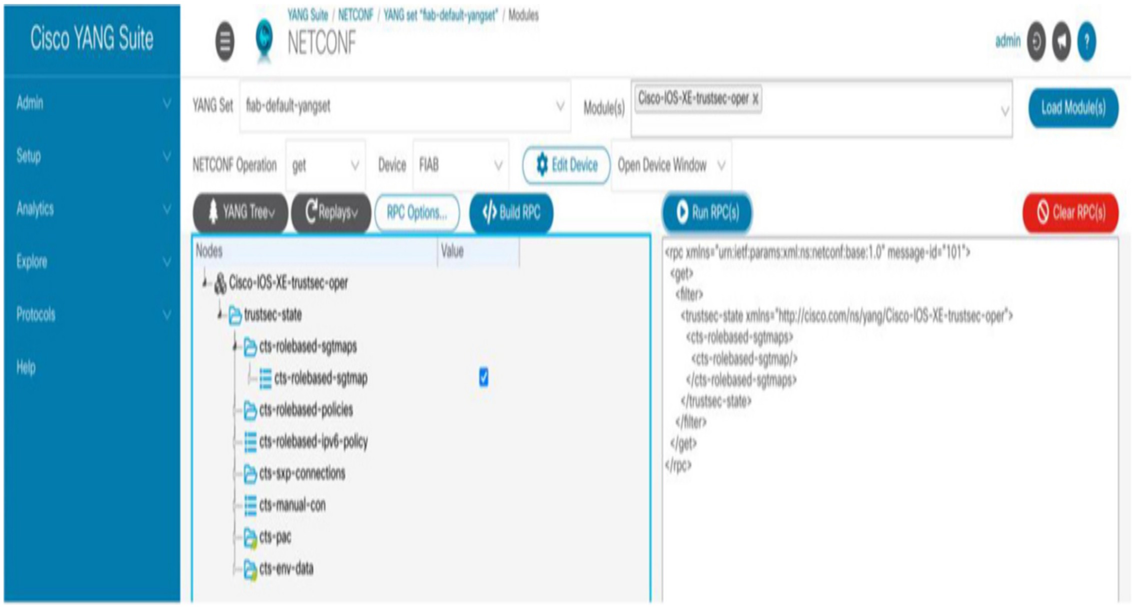
Example 4-3 *SGT-to-Group Mappings IOS XE*

```
C2-Border-1#show cts role-based sgt-map vrf ENT all
Active IPv4-SGT Bindings Information


IP Address              SGT     Source
```

```
============================================
0.0.0.0/1               999     CLI
10.17.122.1             44190   CLI
10.20.14.34             18      CLI
10.20.20.1              2       INTERNAL
10.20.20.8              20      LISP
10.20.20.20             20      LISP
10.86.204.1             2       INTERNAL
10.86.204.13            2       INTERNAL
128.0.0.0/1             999     CLI

IP-SGT Active Bindings Summary
============================================
Total number of CLI      bindings = 4
Total number of LISP     bindings = 2
Total number of INTERNAL bindings = 3
Total number of active   bindings = 9

Active IPv6-SGT Bindings Information

IP Address                                  SGT     Source
================================================================
2001:DB8:C1:1::1                              2       INTERNAL
2001:DB8:C1:1::D                              2       INTERNAL
2001:DB8:C2:330::1                            2       INTERNAL
2001:DB8:C2:330:100:A08B:A4F0:C468            20      LISP
2001:DB8:C2:330:A4A:D8F1:EE39:283B            20      LISP
2001:DB8:C2:330:354B:5D6:1895:6797            255     LISP
2001:DB8:C2:330:5AD0:D27B:1809:DE93           255     LISP
2001:DB8:C2:330:61B6:3117:BC13:ABE2           20      LISP
2001:DB8:C2:330:E937:AFAF:77A2:EDA2           20      LISP
2001:17:122::1                                44190   CLI
```

```
IP-SGT Active Bindings Summary
============================================
Total number of CLI      bindings = 1
Total number of LISP     bindings = 6
Total number of INTERNAL bindings = 3
```

Network operator and security teams that prefer a programmability-based approach can leverage NETCONF, using the popular Cisco YANG Suite tool shown in Figure 4-9 and Figure 4-10 to retrieve the data in an XML-based format that provides a more simplistic means to interface with automation systems and frameworks.



YANG Set fiab-default-yangset

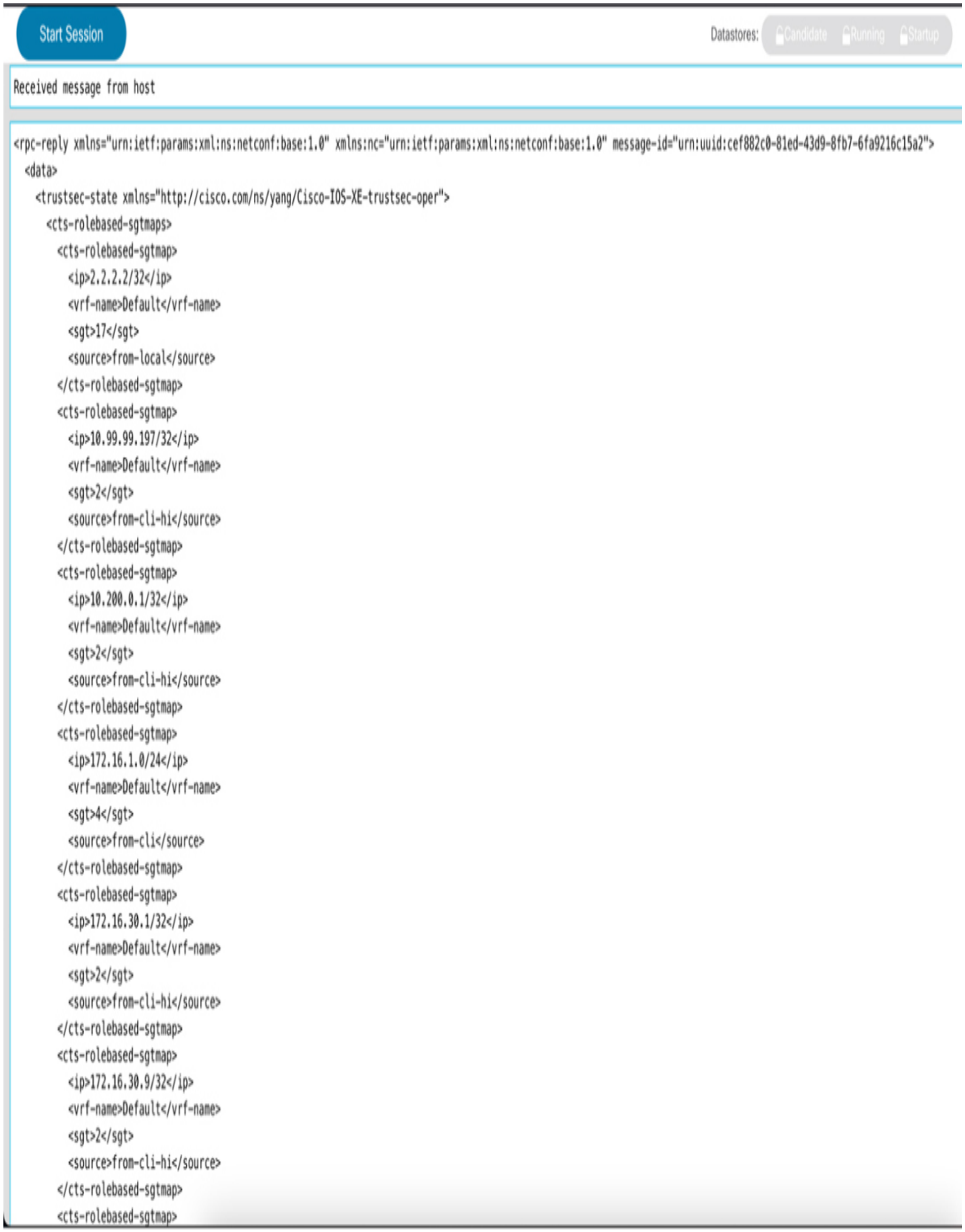
Module(s) Cisco-IOS-XE-trustsec-oper

NETCONF Operation get

Device FIAB

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
 <get>
  <filter>
   <trustsec-state xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-trustsec-oper">
    <cts-rolebased-sgtmaps>
     <cts-rolebased-sgtmap/>
    </cts-rolebased-sgtmaps>
   </trustsec-state>
  </filter>
 </get>
</rpc>
```

Figure 4-9 *Cisco YANG Suite Remote Procedure Call*



Received message from host

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="urn:uuid:cef882c0-81ed-43d9-8fb7-6fa9216c15a2">
  <data>
    <trustsec-state xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-trustsec-oper">
      <cts-rolebased-sgtmaps>
        <cts-rolebased-sgtmap>
          <ip>2.2.2.2/32</ip>
          <vrf-name>Default</vrf-name>
          <sgt>17</sgt>
          <source>from-local</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
          <ip>10.99.99.197/32</ip>
          <vrf-name>Default</vrf-name>
          <sgt>2</sgt>
          <source>from-cli-hi</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
          <ip>10.200.0.1/32</ip>
          <vrf-name>Default</vrf-name>
          <sgt>2</sgt>
          <source>from-cli-hi</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
          <ip>172.16.1.0/24</ip>
          <vrf-name>Default</vrf-name>
          <sgt>4</sgt>
          <source>from-cli</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
          <ip>172.16.30.1/32</ip>
          <vrf-name>Default</vrf-name>
          <sgt>2</sgt>
          <source>from-cli-hi</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
          <ip>172.16.30.9/32</ip>
          <vrf-name>Default</vrf-name>
          <sgt>2</sgt>
          <source>from-cli-hi</source>
        </cts-rolebased-sgtmap>
        <cts-rolebased-sgtmap>
```

Figure 4-10 *Retrieved SGT Mappings in XML Format*

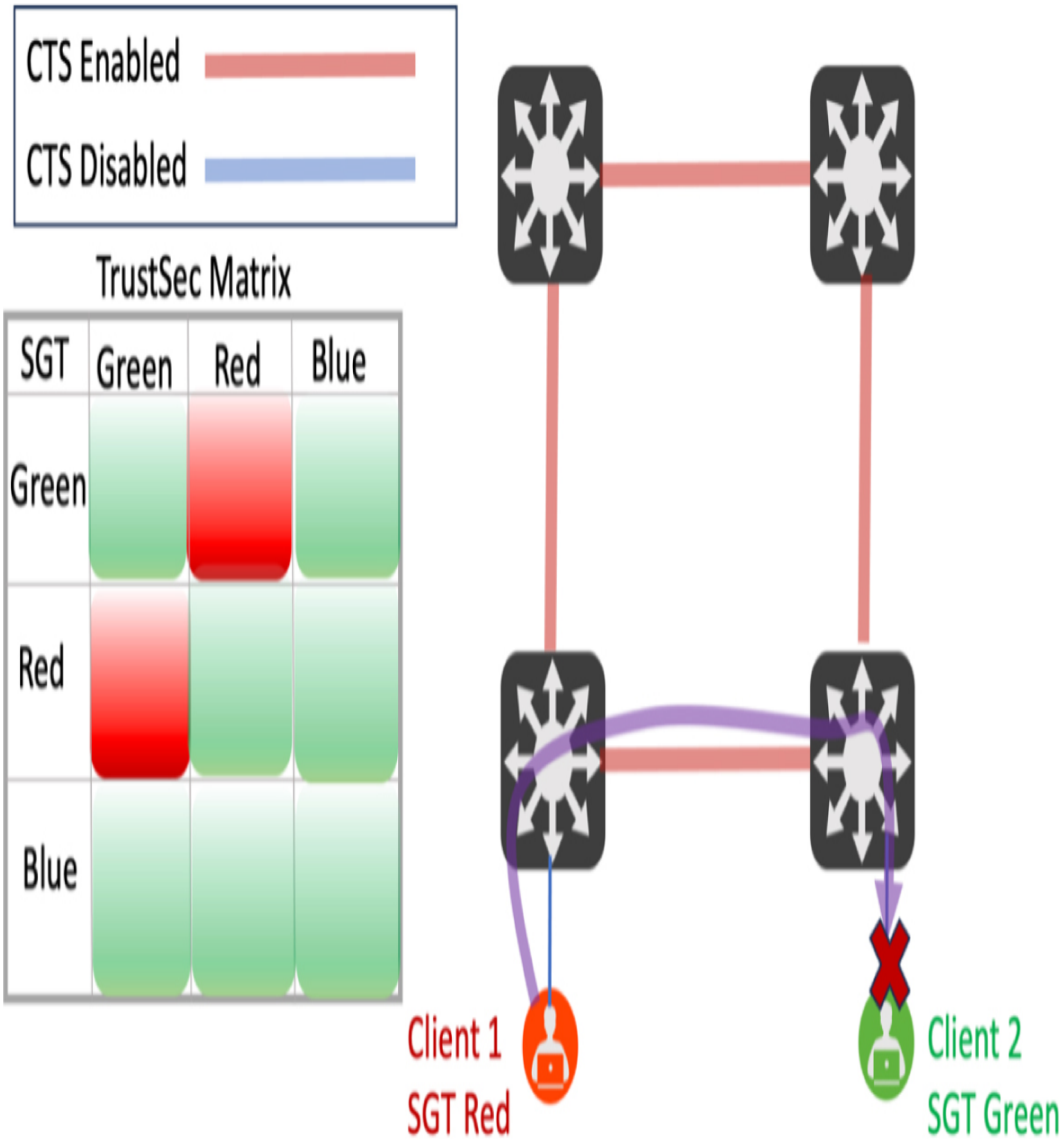
Methods of TrustSec Transport

So far in this chapter, we have described the concepts associated with microsegmentation, including how an operator can choose to view and/or configure policies associated with the technology. With this foundation, it is important to understand how the security group tag information is maintained from one side of an IP network to the other.

With the deployment of TrustSec, two key options are available for how the tag information is propagated across the network architecture, which is applicable to both campus and WAN deployments:

- Maintaining the SGT via the control plane
- Maintaining the SGT via the data plane

It is generally considered that the most robust means of carrying a security group tag within a network is via the data plane, meaning that it is encapsulated as an extra header field in the IP packet and is transported hop by hop from source to destination within the network, as depicted in Figure 4-11.



TrustSec Matrix

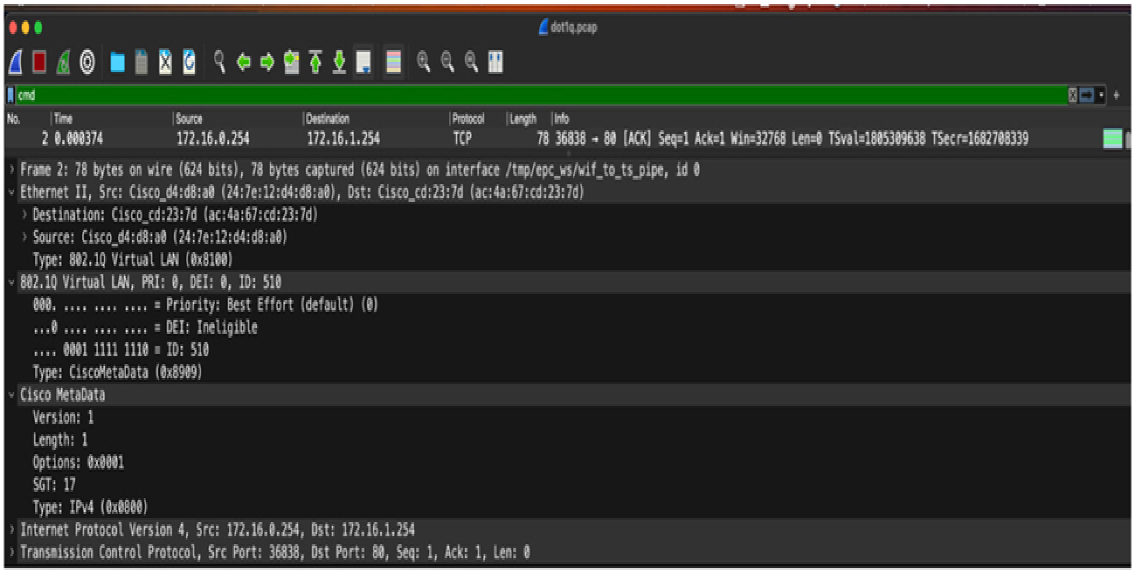
SGT	Green	Red	Blue
Green	🟩	🟥	🟩
Red	🟥	🟩	🟩
Blue	🟩	🟩	🟩

Figure 4-11 *Data Plane Transport of TrustSec Security Group Tags*

An alternative option that network and security teams can pursue is carrying the SGT information in a similar manner to the BGP external routing protocol, whereby a protocol is transmitting the SGT-to-IP binding information to relevant nodes within the network.

CTS Inline Tagging

Inline tagging with TrustSec refers to the ability to perform an encapsulation of the Ethernet frame with a 2-byte Cisco metadata header, which is inserted between the Ethernet or 802.1Q VLAN header of a packet. This deployment method ensures that each packet that is being carried through the network maintains its tag information for transmission with each relevant data payload, as shown in the Wireshark output in Figure 4-12.

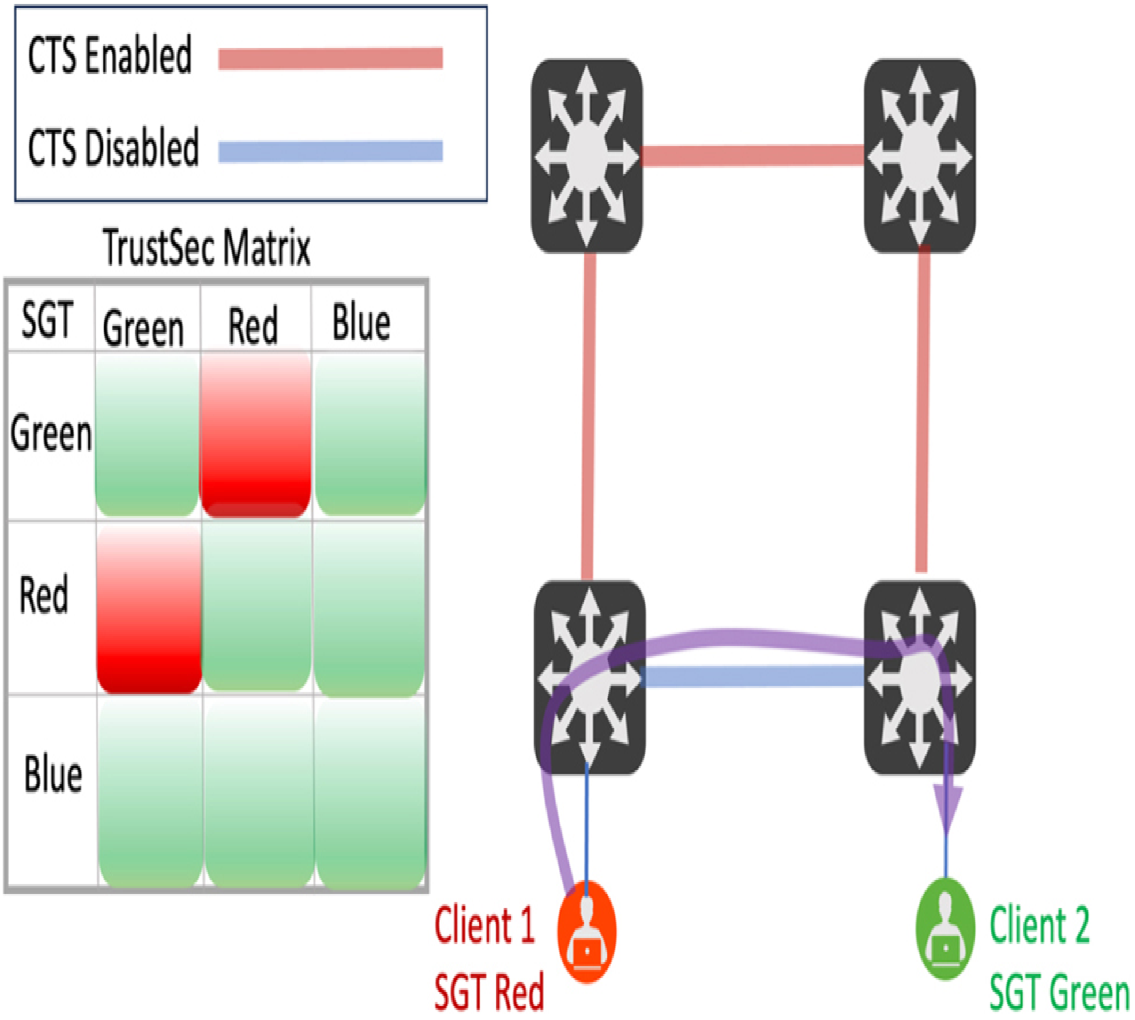


```
No.    |Time          |Source          |Destination      |Protocol  |Length  |Info
     2 0.000374        172.16.0.254      172.16.1.254      TCP          78 36838 → 80 [ACK] Seq=1 Ack=1 Win=32768 Len=0 TSval=1805309638 TSecr=1682708339

› Frame 2: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0
⌄ Ethernet II, Src: Cisco_d4:d8:a0 (24:7e:12:d4:d8:a0), Dst: Cisco_cd:23:7d (ac:4a:67:cd:23:7d)
  › Destination: Cisco_cd:23:7d (ac:4a:67:cd:23:7d)
  › Source: Cisco_d4:d8:a0 (24:7e:12:d4:d8:a0)
    Type: 802.1Q Virtual LAN (0x8100)
⌄ 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 510
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0001 1111 1110 = ID: 510
    Type: CiscoMetaData (0x8909)
⌄ Cisco MetaData
    Version: 1
    Length: 1
    Options: 0x0001
    SGT: 17
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 172.16.0.254, Dst: 172.16.1.254
› Transmission Control Protocol, Src Port: 36838, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
```

Figure 4-12 *Cisco Metadata Header Wireshark Capture—Inline Tagging*

This concept is not new to the deployment of TrustSec. Over the years, many protocols achieved their functions by being inserted in between header fields. One such example was the Class of Service (CoS) field, which was used for quality of service (QoS) in switched networks. CoS enabled the operator to ensure that QoS values were transported over network interfaces when configured. One of the key challenges that often occurred with network operators who configured CoS was that when parallel links in their switching networks were added or upgraded links were deployed, implementation teams would forget to configure the CoS configurations on the new interface, leading to the important quality of service markings being lost when being transmitted over the new or upgraded links.

Much like the deployment of CoS in legacy switching networks, on Cisco IOS XE platforms, the configuration of inline tagging takes place on a per interface basis. This means that the challenges that were observed in legacy networks with the loss of CoS markings can also apply to Cisco TrustSec domains, in scenarios where an operator may forget to configure the inline tagging configurations on the interface, as shown in the Figure 4-13.



SGT	Green	Red	Blue
Green	🟩	🟥	🟩
Red	🟥	🟩	🟩
Blue	🟩	🟩	🟩

Figure 4-13 *Loss of SGT Due to Missing Configuration on Link*

To configure inline tagging in a network deployment, you can use the **cts manual** command on IOS XE, as can be seen in Example 4-4.

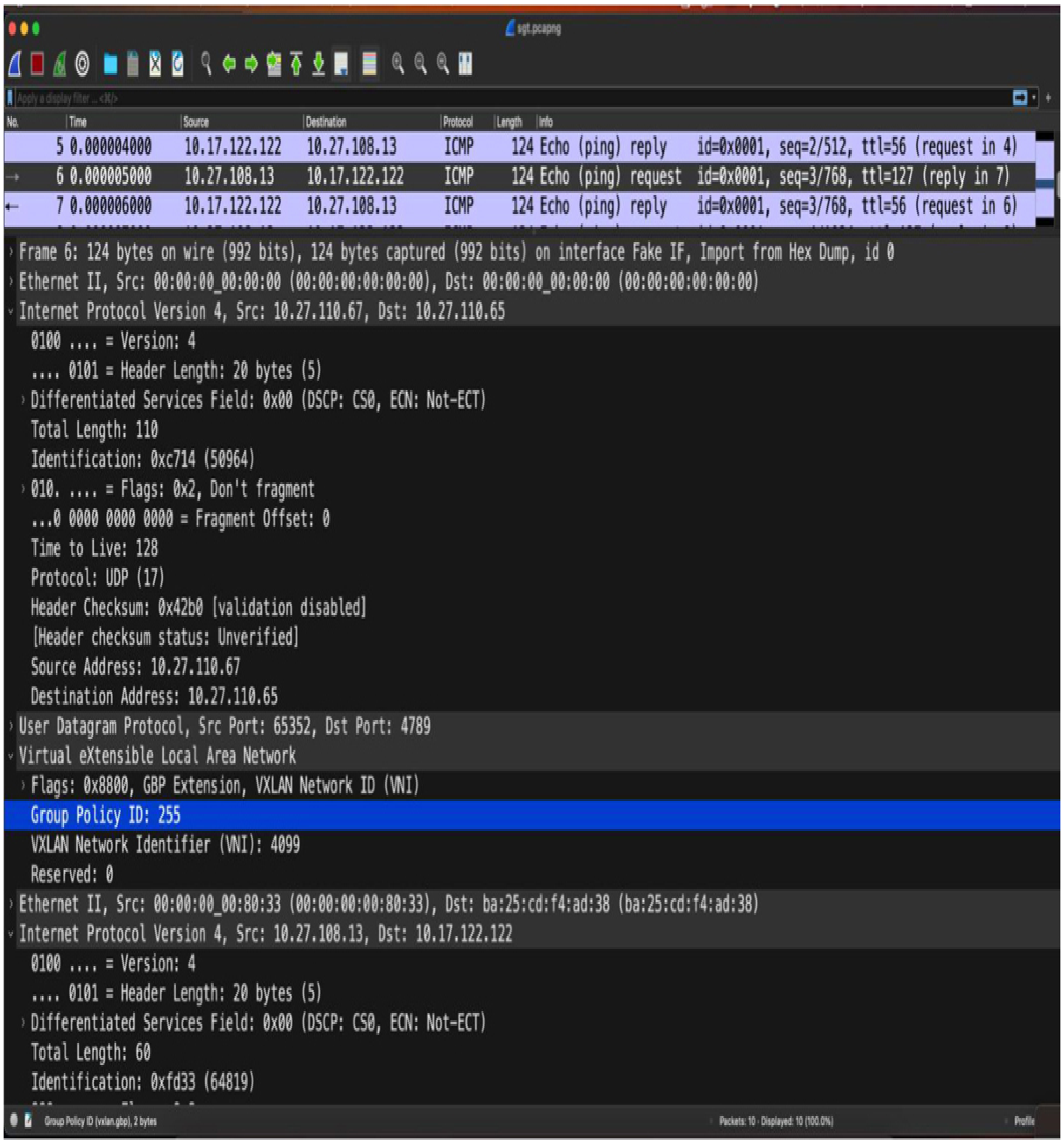
Example 4-4 *TrustSec Interface Configuration in IOS XE*

```
Branch-FIAB#show run int gig 1/0/21
Building configuration...

Current configuration : 444 bytes
!
```

```
interface GigabitEthernet1/0/21
 description HQ_cEdge-0_2_0
 switchport mode trunk
 ip flow monitor dnacmonitor input
 ip flow monitor dnacmonitor_dns input
 ip flow monitor dnacmonitor output
 ip flow monitor dnacmonitor_dns output
 ipv6 flow monitor dnacmonitor_v6 input
 ipv6 flow monitor dnacmonitor_dns_v6 input
 ipv6 flow monitor dnacmonitor_v6 output
 ipv6 flow monitor dnacmonitor_dns_v6 output
 cts manual
  policy static sgt 2
end

Branch-FIAB#
```

VXLAN Encapsulation

Within Software-Defined Access and Cisco EVPN deployments, VXLAN is used as a data plane method to carry SGT information used to maintain microsegmentation. VXLAN is leveraged as the protocol to not only encapsulate data plane IPv4 or IPv6 traffic from client and server devices that are located in the fabric, but also having a Group Policy field for the SGT and VXLAN Network Identifier field for VRF information, which can be seen in Figure 4-14. VXLAN also maintains inner-to-outer QoS mappings by copying DSCP values from the inner packet to the outer header, thereby allowing the mappings across the underlay for its respective client traffic.


sgt.pcapng
Apply a display filter ... <⌘/>
Group Policy ID (vxlan.gbp), 2 bytes
Packets: 10 · Displayed: 10 (100.0%)
Profile

```
No.    Time          Source          Destination     Protocol  Length Info
     5 0.000004000   10.17.122.122   10.27.108.13    ICMP      124 Echo (ping) reply    id=0x0001, seq=2/512, ttl=56 (request in 4)
→    6 0.000005000   10.27.108.13    10.17.122.122   ICMP      124 Echo (ping) request  id=0x0001, seq=3/768, ttl=127 (reply in 7)
←    7 0.000006000   10.17.122.122   10.27.108.13    ICMP      124 Echo (ping) reply    id=0x0001, seq=3/768, ttl=56 (request in 6)

› Frame 6: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface Fake IF, Import from Hex Dump, id 0
› Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
˅ Internet Protocol Version 4, Src: 10.27.110.67, Dst: 10.27.110.65
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 110
    Identification: 0xc714 (50964)
  › 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x42b0 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.27.110.67
    Destination Address: 10.27.110.65
› User Datagram Protocol, Src Port: 65352, Dst Port: 4789
˅ Virtual eXtensible Local Area Network
  › Flags: 0x8800, GBP Extension, VXLAN Network ID (VNI)
    Group Policy ID: 255
    VXLAN Network Identifier (VNI): 4099
    Reserved: 0
› Ethernet II, Src: 00:00:00_00:80:33 (00:00:00:00:80:33), Dst: ba:25:cd:f4:ad:38 (ba:25:cd:f4:ad:38)
˅ Internet Protocol Version 4, Src: 10.27.108.13, Dst: 10.17.122.122
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xfd33 (64819)
```

Figure 4-14 *VXLAN Mappings for Macro- and Microsegmentation*

GRE Encapsulation

The generic routing encapsulation protocol, commonly referred to as GRE, is another possibility for carrying Cisco TrustSec metadata on the data plane. Much like VXLAN, it provides the network operator a means to transport traffic over an underlay network that may not be able to adequately support Cisco TrustSec.

Although this option can be very useful for deployments, not every platform can support this capability, even if it supports the GRE protocol. At the time of writing, some of the ISR, ASR, and C8K routing platforms could provide this capability, but it was not yet available in the Catalyst 9000 Series of devices.

IPsec Encapsulation

Various Cisco platforms can carry TrustSec data within IPsec payloads, providing the ability to securely carry TrustSec information end to end across an insecure IP network or even the public Internet. On supporting Cisco platforms, the tags can be carried through the negotiation with IKEv2:

- DMVPN
- Dynamic virtual tunnel interface (dVTI)
- GRE with tunnel protection
- Site-to-site VPNs
- Static crypto maps
- Static virtual tunnel interface (sVTI)

In addition to the deployments listed, the Cisco Meraki SD-WAN and Cisco Catalyst SD-WAN solutions also can maintain the TrustSec tag information over the wide area network.

Given the simplicity that is provided by the network orchestrators in terms of provisioning of the underlying network, the most common and typical solutions that are seen in customer deployments today are TrustSec being deployed on the network infrastructure via Cisco network orchestrators. While very large customer networks have also pursued deployment via other methods, the simplicity awarded through the configuration being performed by orchestrator-based deployments has lowered the entry point for many customers on their journey toward the use of microsegmentation as a component of their zero trust estate.

Control Plane TrustSec Transport

In some scenarios it is simply not possible to maintain SGT information end to end across an IP network. These constraints and challenges may arise when there is a third-party SD-WAN or a managed MPLS, service provider, or WAN in between TrustSec-capable sites or hyperscaler cloud locations.

For these sorts of deployment scenarios, fortunately, a number of tools can be used to ensure that TrustSec mappings can still be applied, allowing end-to-end network enforcement to be achieved.

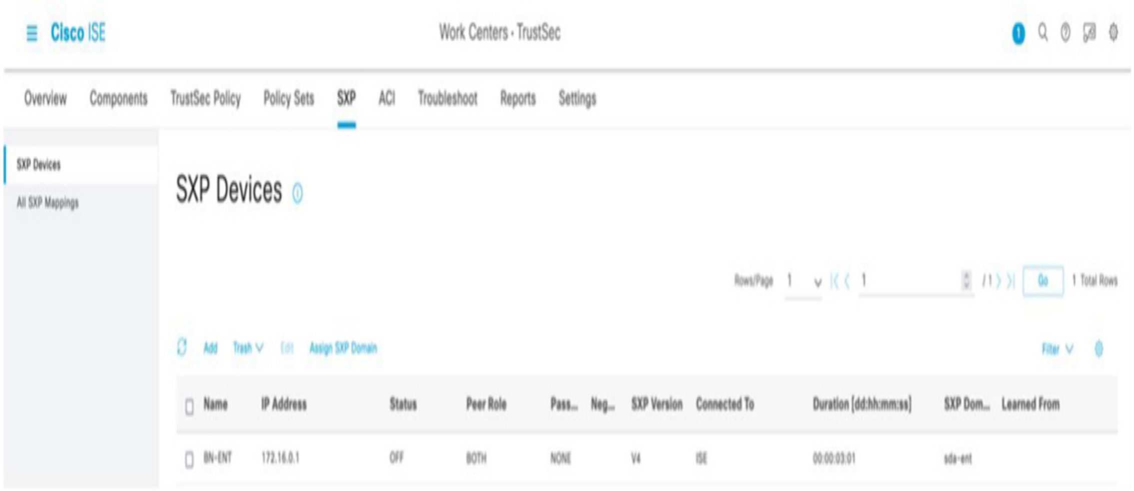
SXP

Scalable Group Tag Exchange Protocol, or SXP, was submitted as draft-smith-kandula-sxp-00 to the IETF in July 2014 as a means to be able to maintain and propagate and learn security group tag information between network systems. This draft expanded over the years, leading to newer iterations of the protocol, with SXPv4 being adopted within customer networks for the transmission of TrustSec information in various domains.

While use of the protocol was common, a number of caveats were apparent with its usage:

- SXPv4 required a peering per VRF.
- A large amount of memory was needed to maintain bindings.

In October 2022, a newer iteration of SXP was made available (SXPv5), including some further enhancements. The new iteration included the addition of a 2-byte VLAN identifier field, a 32-byte VRF name field, and version negotiation capabilities. As a result, fewer SXP peerings are required to transmit SGT mappings, thus lowering the complexity associated with the configuration of the protocol within network domains. Figure 4-15 shows an overview of configured SXP peerings in ISE.



SXP Devices

Name	IP Address	Status	Peer Role	Pass...	Neg...	SXP Version	Connected To	Duration [dd:hh:mm:ss]	SXP Dom...	Learned From
BN-ENT	172.16.0.1	OFF	BOTH	NONE		V4	ISE	00:00:03:01	sda-ent	

Figure 4-15 *SXP Peerings in ISE*

LISP

From IOS XE version 17.8.x and higher, SGT carriage over LISP was introduced into the Cisco Catalyst 9000 platform software. The introduction of this capability provided more holistic ways of ensuring that SGT information could be mapped and maintained to hosts within a routing domain using the LISP protocol, such as Software-Defined Access. Initially, this capability was introduced to augment the mappings of highly dynamic hosts, such as devices which existed within a wireless environment that had a high likelihood of rapid mobility requirements. By using LISP, you could execute a means to remap hosts that roamed from one edge switch (ITR/ETR) to another with relative ease.

In later code from 17.12.x, this capability was further enriched to enable LISP to carry SGT information over a nonsupporting network, even over the public Internet.

At the time of writing, although this capability is available in the Cisco IOS XE code and is exercised within a site for the wireless bridge virtual machine feature to support and facilitate roaming of bridged VM clients, it has yet to be automated as part of Catalyst Center’s execution flow for scenarios beyond this use case.

Static

Within IOS XE software, it is possible to create static binding configurations of the respective IP addresses that may exist in the global table or within a VRF and corresponding SGT mapping. Such static allocations provide a means for client devices that may not have an allocation over radius or be properly allocated with a subnetwide mapping to ensure that the right allocation is indeed in place.

When we consider scenarios that involve static versus pool mapping, the more specific allocation of the static IP to binding mapping will win over scenarios where there is a broader match.

The configuration for static entries is shown in Example 4-5.

Example 4-5 *Configuring Static Bindings on IOS XE*

```
Edge#show run | i cts role-based sgt-map
cts role-based sgt-map vrf ENT 8.8.4.4 sgt 8844
cts role-based sgt-map vrf ENT 8.8.8.8 sgt 8888
Edge#
```

As you can see in the entries with the source listed as CLI in Example 4-6, this represents bindings of a static nature that were manually configured by an operator.

Example 4-6 *Corresponding Static Binding in SGT Mapping Table*

```
Edge#show cts role-based sgt-map vrf ENT all
Active IPv4-SGT Bindings Information

IP Address              SGT     Source
============================================
1.1.1.1                 17      LOCAL
8.8.4.4                 8844    CLI <<< Static Binding
```

```
8.8.8.8                 8888    CLI <<< Static Binding
172.16.0.1              2       INTERNAL
```

SGT Priority Order

The IP-to-SGT mapping table has a priority order that it traverses to identify which source of TrustSec information should be prioritized for use. For network operators who are familiar with the function of administrative distance in routing protocols, the mapping table for the use of TrustSec is similar.

In scenarios where multiple SGT sources are mapping to the same IPv4 or IPv6 prefix at the same time, the use of the mapping table allows the tie-breaking decision to take place.

The priority mapping ranges from lowest to highest are as follows, with the lowest value representing the best choice:

1. VLAN
2. CLI
3. L3IF
4. LISP_REMOTE_HOST
5. LISP_LOCAL_HOST
6. OMP
7. SXP
8. ARP
9. LOCAL
10. CACHING
11. INTERNAL

In scenarios where multiple methods to map SGTs to IP addresses are in use, it is important for operators to be mindful of the preceding list order to ensure that the desired allocations and outcomes are achieved.

Secure Service Insertion

Microsegmentation has the capacity to provide significant benefits in achieving per client or per device segmentation and separation without unnecessarily tromboning or hairpinning traffic. For the majority of environments, sending traffic all the way up to a northbound set of firewalls or fusion devices that would allow route leaking to take place is typically something that is avoided when following best practices, unless there is a need for an audit trail to exist, as described earlier in this chapter. For some domains, however, particularly manufacturing and industrial, compliance reasons force them to ensure that certain communications are audited and protocoled and, as such, must traverse a firewall. This activity is achievable when leveraging VRF-based separation policies, but performing such actions rapidly becomes challenging when attempting to do it at scale.

Some Cisco customers need to have VRF-like segmentation for each and every new IoT project. If we’re talking about 5 to 10 VRFs, this is not the end of the world; but if we’re considering thousands, many roadblocks come into play, ranging from platform limits associated with TCAM utilization to complexity associated with IP address planning, naming conventions, routing policy, and VLAN ID allocations.

Fortunately, through the use of the secure service insertion functionality, there is another way to achieve such use cases in network environments, as shown in Figure 4-16.

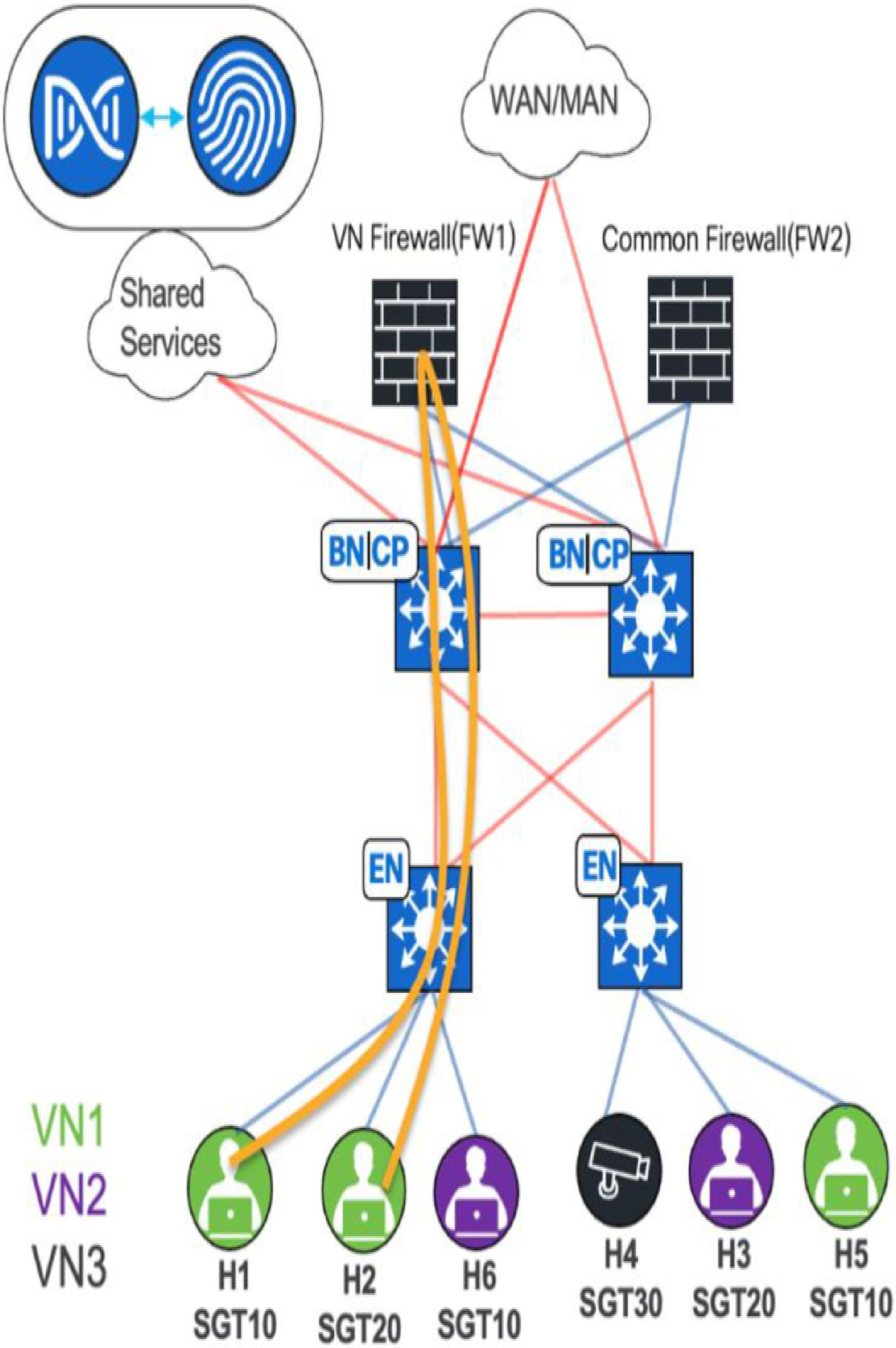


Figure 4-16 *Secure Service Insertion Traffic Flow*

This functionality allows scenarios whereby two devices within the same virtual routing and forwarding instance are able to be selectively sent to a firewall for inspection, based on a configurable ruleset. This approach allows for a significantly higher scale of segmentation deployment, with the dynamic ability to ensure that requisite auditability is available for flows where this may be required.

Through the use of secure service insertion, operators can gain the best of both worlds—having the classic capabilities that are awarded through the use of microsegmentation, while at the same time being able to perform on-demand redirection to security appliances such as firewalls for stateful inspection.

LAN-to-Cloud Microsegmentation

Challenges associated with the ability to maintain end-to-end microsegmentation have arisen in many customer domains over the years. However, the value that the technology can bring tends to outweigh the potential complexities that can arise. In particular, the ability to prevent insecure peer to peer communication, sometimes referred to as lateral traversal. The fact that the enforcement of the policy takes place on the remote edge and between disparate domains can, under certain circumstances, be problematic. The challenges that may have arisen in the past tended to come down to the right planning and scale.

For example, consider a northbound data center that has thousands of workloads that are not correctly mapped to TrustSec within their domain. To achieve the requisite enforcement, SXP was often leveraged, allowing an operator to manually create mappings of SGTs to IP addresses. While this approach was functional, it often resulted in significant numbers of bindings having to be maintained in locations that were many hops away from the service to enforce against, resulting in a higher scale requirement on the intermediate platform tasked with the subsequent enforcement.

With the introduction of common policy in ISE 3.4, these challenges have begun to be addressed, specifically through a means of harmonizing rulesets that are leveraged within different domains as a result of a mapping the user identity. In this way, architectures such as ACI or workloads that may exist northbound in the cloud using a hyperscaler like AWS, Azure, and GCP (as shown in Figure 4-17) can use ISE through PXGRID to share their context.

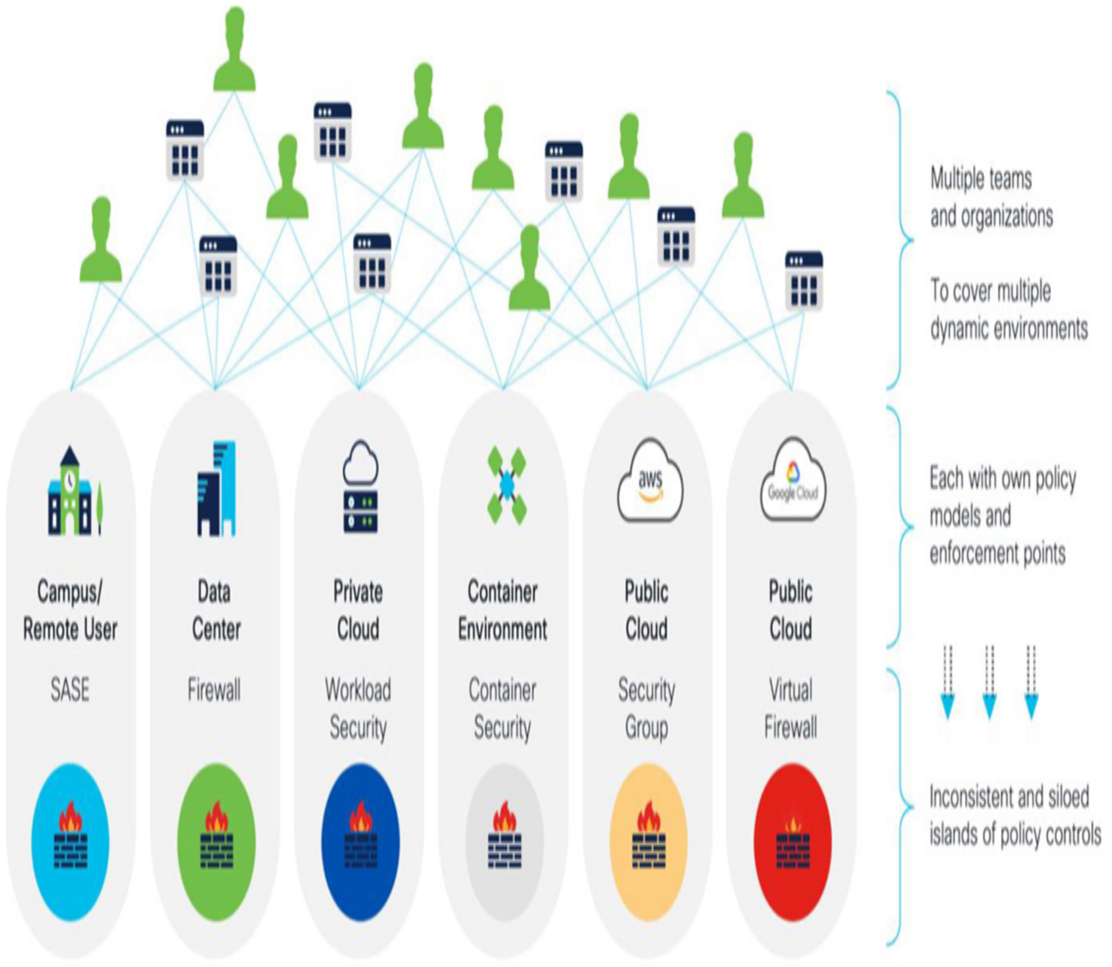


Figure 4-17 *Common Policy Allows for Broader Deployment of Microsegmentation*

Considering that, in modern networks, over 60 percent of traffic is destined toward a cloud service or data center, the addition of this capability simplifies the end-to-end standardization of policy needed to ensure continuity within an architecture. This approach can significantly lower the complexity associated with such a deployment.

Summary

In this chapter, we described macro- and microsegmentation, including their importance in providing the capability to enforce zero trust security principles by limiting communications to resources to only systems that require them. These capabilities represent an important foundation in enabling security, network, and cloud teams to apply the right security within their respective organizations.

Part 2: Network Automation Capabilities in Software Defined Architectures

Chapter 5. DHCP and Dynamic Addressing Concepts

In this chapter, you will learn about the following:

- Dynamic addressing concepts in IPv4
- The zero trust approach to dynamic addressing
- IPv6 addressing and assignment methods
- IPv6 first hop security features

Introduction to Dynamic Addressing

An *IP address* is a unique identifier for a device on a local network or the Internet. It’s a part of the TCP/IP suite of protocols. An IP address serves two specific functions: to identify the device on the network and to indicate its location within the larger network. This functionality becomes important when we need to route communication packets between two devices across several routers. However, there are new concepts, like the Locator ID Separation Protocol (LISP), that untie the bundling of location and identification from the IP address. LISP forms the control plane for the Cisco SD-Access solution. In this chapter, we will focus on traditional IP address mechanisms with common security and resiliency approaches.

In the early days of networking, IP addresses were manually assigned to devices. This meant that IT administrators had to configure each device with a unique IP address by hand. This process was time-consuming and prone to errors like the following:

- Static IP address assignment required careful planning to ensure no two devices were assigned the same IP address, which would disrupt network communication.

- This operation was not scalable. As networks grew, issues like assigning duplicate addresses or updating the addresses when devices moved to a new location were common.

To solve the problems of manual IP management, dynamic addressing was introduced. Reverse Address Resolution Protocol (RARP) was the first effort at solving this issue, but it was severely limited because it was not able to assign the default gateway and name servers. Both parameters were critical for network access. In a RARP process,

- The client sends a broadcast RARP on the network with its MAC address.
- A RARP server receives the request and looks up the static mapping of MAC-IP binding in its local database.
- The RARP server then sends the response to the requesting device with its IP address. In this case, the destination address is 0.0.0.0, and the IP assigned is sent as payload.

RARP was not scalable due to the MAC-IP requirement of static mapping. Bootstrap Protocol (BOOTP) was the first protocol-based attempt to automate the process of IP address assignment. It was defined in RFC 951. The idea was to assign the IP details to the diskless devices because they did not have the local disk to store their static configuration. It was a simple protocol where, after booting up, the client sent out a BOOTP broadcast request to the network. This request was then picked up by the server listening on UDP port 67. This approach solved the problem of assigning default gateway and name server assignments. It also supported providing additional configuration options like the location of the boot file, which was critical for diskless clients but still required static mapping of MAC-IP to be stored. The primary intent of BOOTP was to assist the diskless client's boot process and not to assign the IP addresses. BOOTP supported the provision to assign an IP address if the initial request from the client did not contain an IP address. BOOTP also allowed clients to send optional parameters like the name of the server that they wished to get a reply from and a generic boot filename, such as "Unix." Figure 5-1 shows this process.

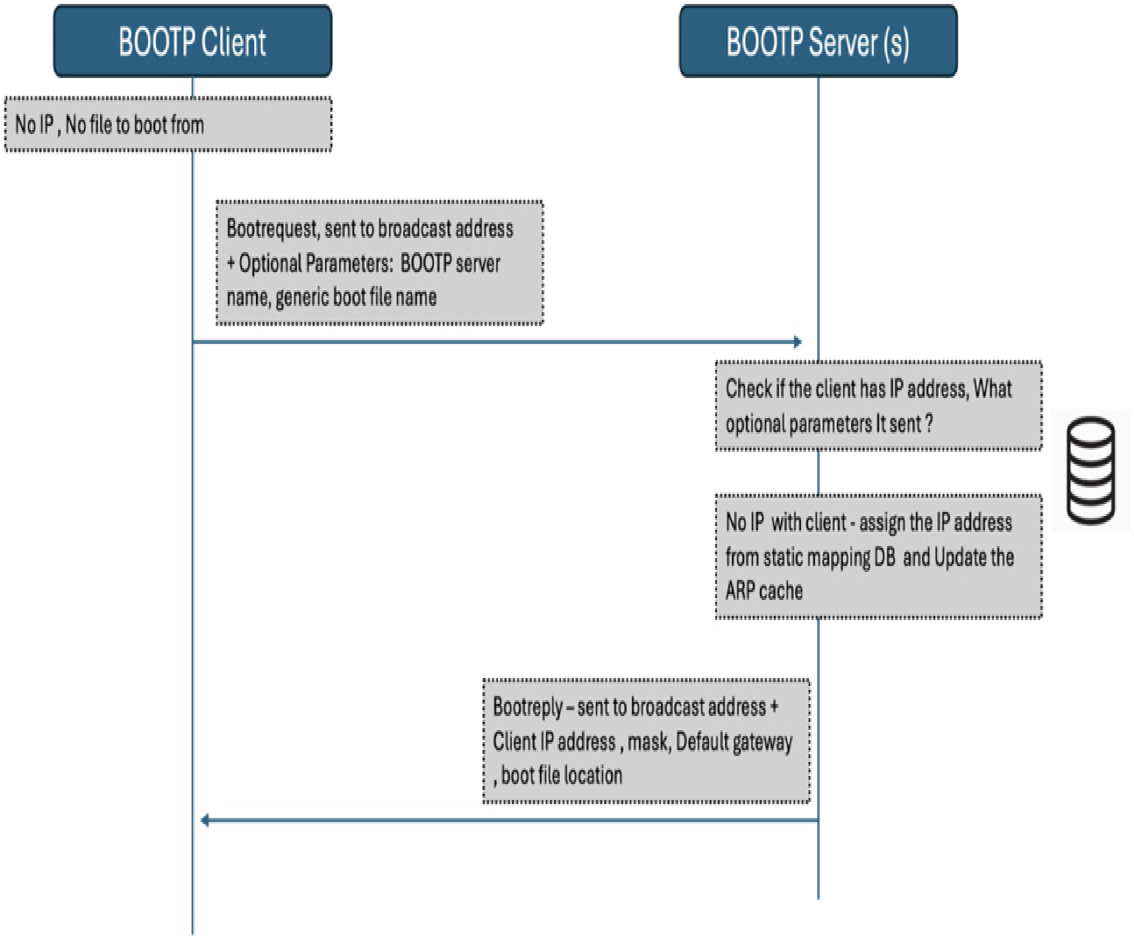

BOOTP Client
BOOTP Server (s)
No IP , No file to boot from
Bootrequest, sent to broadcast address + Optional Parameters: BOOTP server name, generic boot file name
Check if the client has IP address, What optional parameters It sent ?
No IP with client - assign the IP address from static mapping DB and Update the ARP cache
Bootreply – sent to broadcast address + Client IP address , mask, Default gateway , boot file location

Figure 5-1 *BOOTP Process*

BOOTP was an important step in the evolution of the network configuration protocols. It has set the stage for more advanced protocols like DHCP with additional capabilities. The Dynamic Host Configuration Protocol was first defined as a standards-track protocol in RFC 1531 in October 1993; it was an extension to the Bootstrap Protocol (BOOTP), a network protocol used by a network client to obtain an IP address from a configuration server.

As networks became more complex and new use cases were getting deployed, there was a need for a protocol that was flexible and able to provide additional information to the clients beyond IP addresses. DHCP introduced several additional functionalities and enhancements when compared to the BOOTP:

- **Leasing Mechanism:** DHCP introduced the concept of IP addressing leasing, unlike previous methods where IP addresses were tied to the devices’ MAC addresses. Clients can now lease the IP addresses for a specific duration. This allows efficient reuse of the IP addresses in a network.

- **Automation Reallocation:** DHCP can automatically reassign the addresses that are no longer in use by other clients.
- **DHCP Options:** DHCP can provide comprehensive network configuration to the clients via Options. For example, Option 42 carries details about the NTP server that allows time synchronization across devices, and Option 3 provides the IP address of the default gateway. It is critical for log analysis and processing to have a correct timestamp on the logs.
- **Client Acknowledgment:** With a four-step communication process, including Discover, Offer, Request, and Acknowledgment, there was a mechanism that could confirm the receipt of details from the server and also give flexibility to the client to accept or reject offers.
- **Custom Configuration:** DHCP also supports vendor-specific options, allowing for the distribution of the custom configuration settings. For example, DHCP Option 43 provides a wireless LAN controller IP address to the wireless access points, allowing them to send the connection request. Figure 5-2 shows the typical DHCP flow.

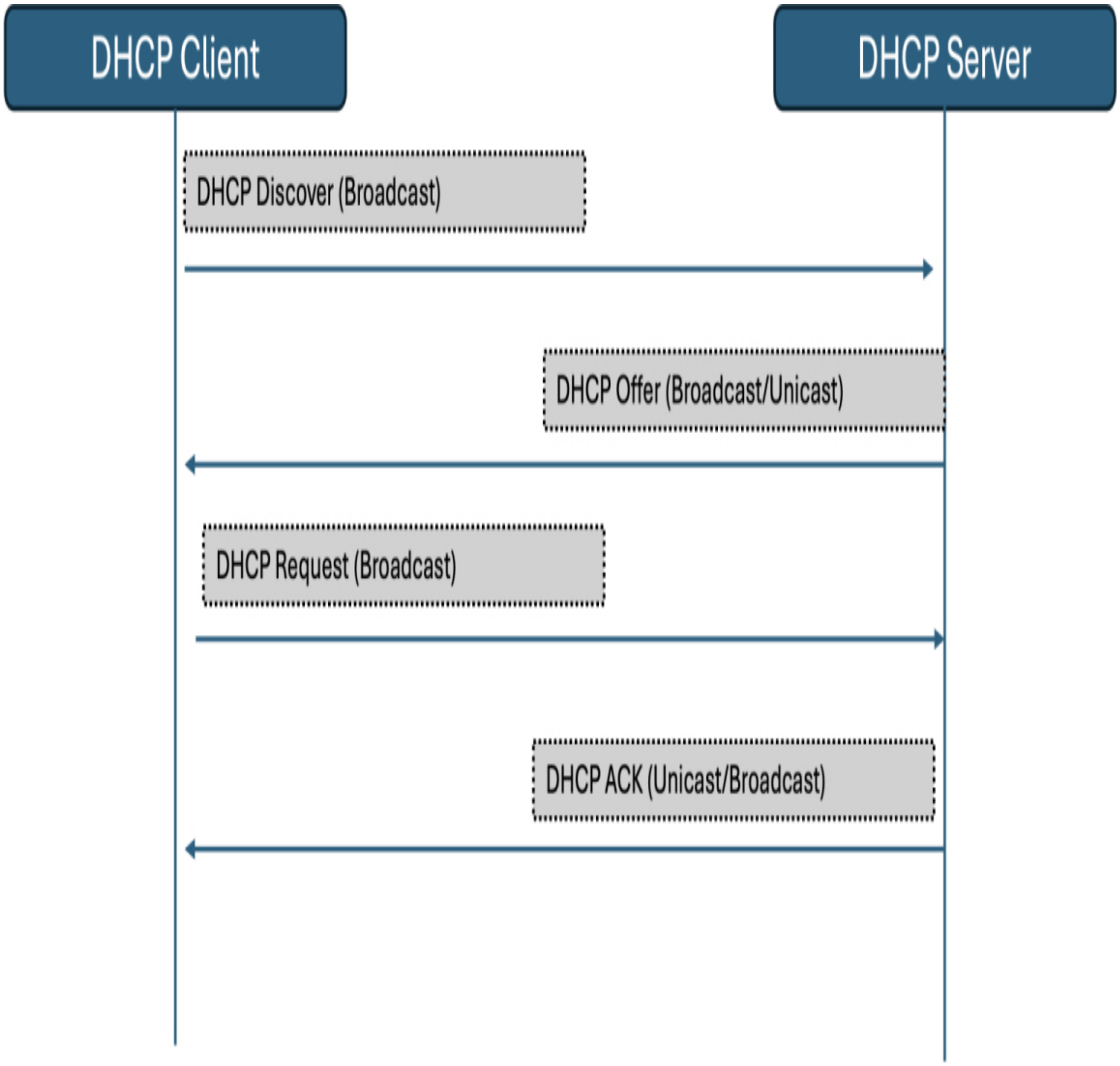

DHCP Client
DHCP Server
DHCP Discover (Broadcast)
DHCP Offer (Broadcast/Unicast)
DHCP Request (Broadcast)
DHCP ACK (Unicast/Broadcast)

Figure 5-2 *DHCP Frame Exchange*

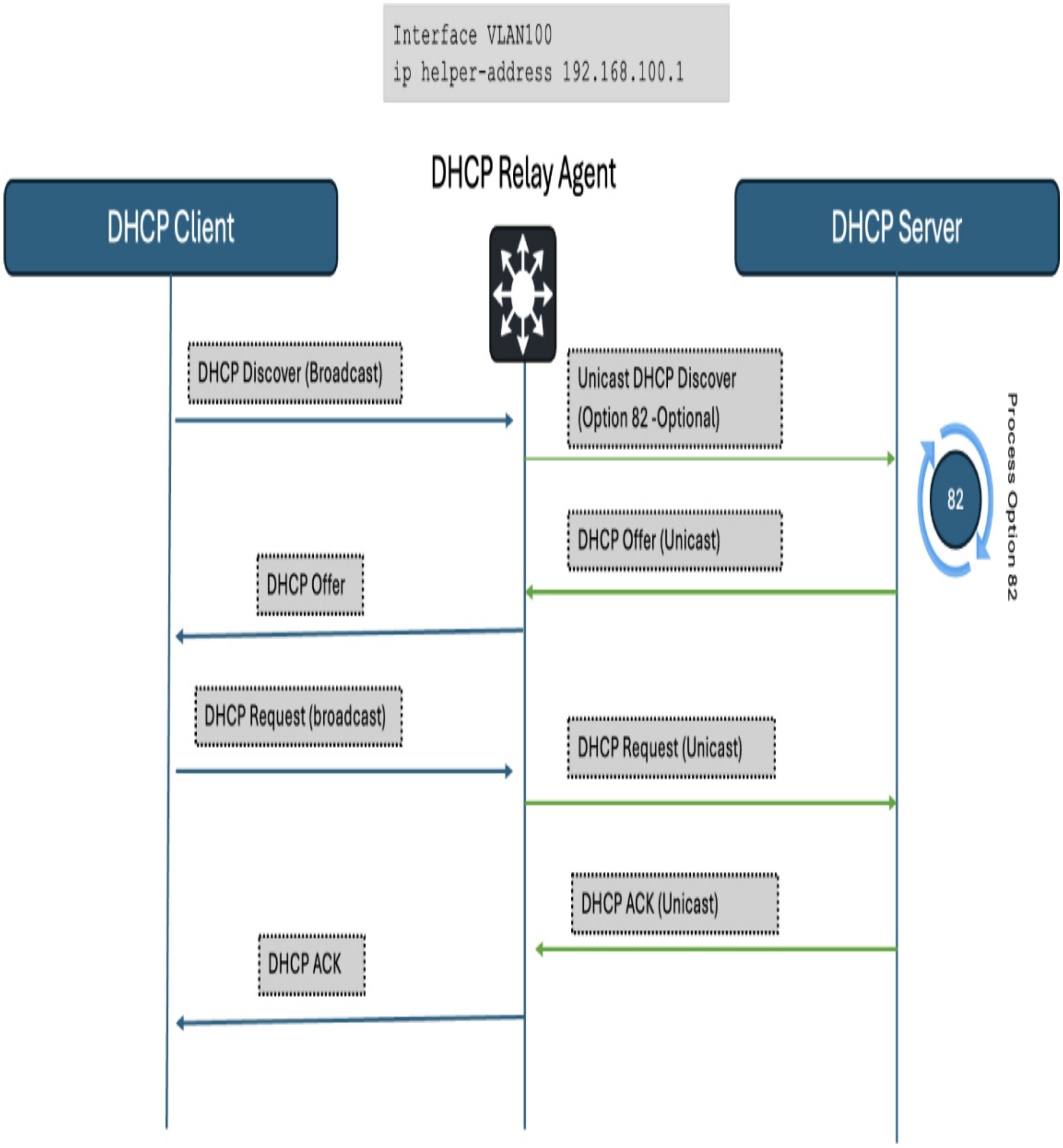
You can see that the DHCP offer is a broadcast, but under certain conditions, this message is sent as unicast:

- If the client has included the previously assigned IP address as part of Option 50, this conveys to the DHCP server that the client has an existing IP address. In this case, the DHCP server sends a unicast DHCP offer to that IP address.
- When the client wants to renew its lease, the DHCP offer is sent as unicast.
- When a relay agent is used, in that case, the DHCP server can send a unicast DHCP offer to the relay agent. You will learn about relay

agents later in this chapter.

- DHCP inform is also sent as unicast. The client sends an inform message to request additional configuration parameters without obtaining an IP address.

As you know, a DHCP discover message is a broadcast message. It means that this packet will reach the DHCP server only if it is part of the same broadcast domain. In a real-world scenario, you will have a central DHCP server in an enterprise service block that is several routers away from the client. To solve this problem, the DHCP relay agent listens for the DHCP broadcast message from the clients on a subnet and forwards them to the DHCP server on another subnet. These relay agents are typically located on a router or Layer 3 switch that connects multiple subnets. Unlike a regular router, when the relay agent receives a DHCP message, it does not forward the same packet. Instead, it creates a new one. It adds the gateway address (the giaddr field of the DHCP packet) and may include extra information like DHCP Option 82. Then it sends this message to the DHCP server. When the server replies, the relay agent removes Option 82 and forwards the response to the client. Figure 5-3 shows the DHCP process with the relay agent.



```
Interface VLAN100
ip helper-address 192.168.100.1
```

Figure 5-3 *DHCP Process with Relay Agent*

From a security point of view, DHCP relay helps in certain ways:

- DHCP relay allows the DHCP server to be centralized in a secure and controlled environment like a data center.
- Having a DHCP server on every subnet increases the attack surface, making them potential targets for network attacks (for example, spoofing or exhaustion attacks).

- DHCP relay can be configured on firewalls/routers. Access control policies can be created to restrict who can send DHCP requests.

- With DHCP relay, since the relay agent is configured to forward requests only to specific, trusted DHCP servers, it reduces the risk of rogue DHCP servers influencing network clients.

- DHCP relay can be integrated with other network security technologies, such as network access control (NAC) systems. For instance, a DHCP relay agent can work alongside a NAC solution to ensure that only authenticated and compliant devices receive network configurations.

Option 82, the DHCP relay agent information option, can be used to provide additional contextual information about the DHCP requests that can be used for enhancing security and improving network management. As shown in Figure 5-3, you can add additional information about relay agent identity and port/interface where the request was received to the DHCP request. DHCP servers can then be configured to assign IP addresses based on this information. Large ISPs use this information to assign IP addresses based on pools mapped with the location of the client. Option 82 typically includes two suboptions:

- **Circuit ID (Suboption 1):** Identifies the specific interface, port, or circuit on the relay agent where the DHCP request was received. This can include VLAN information, physical port numbers, or other identifying data.

- **Remote ID (Suboption 2):** Identifies the relay agent itself, typically by including a unique identifier like the relay agent's MAC address, a configured ID, or some custom parameters.

Adding Option 82 helps in improving the security posture:

- Option 82 adds information about where a DHCP request comes from, such as the specific switch port or VLAN. This ensures that only devices connected to approved locations can receive an IP address, preventing unauthorized devices from joining the network.

- Rogue devices are unauthorized devices trying to connect to the network. With Option 82, the DHCP server can identify and reject requests from unknown or unexpected locations, blocking these devices from getting an IP address.

- Option 82 allows network administrators to see which device was assigned a specific IP address, based on where the request came from. This capability helps in tracking network usage, auditing, and troubleshooting, making it easier to manage the network.

Note

DHCP Option 82 is also used in the Cisco SD-Access Fabric. Fabric Edge, which is an access switch in Cisco SD-Access Fabric, adds its LISP RLOC as remote ID in Option 82 and sets the address as the anycast SVI. For detailed information on the use of DHCP Option 82 with Cisco SD-Access, refer to the SDA solution guide on Cisco.com.

Consider this example: DHCP Option 82 was used for a large service provider’s country-wide 3/4G to Wi-Fi offload use case. The service provider deployed thousands of access points across the country to ease the congestion on the 3/4G network, especially in crowded places such as malls and airports. The use case was to assign addresses in a specific fashion aligned to the region. We used DHCP Option Required (making sure the client must get an IP address from DHCP via Cisco WLC) and DHCP Option 82, supplying AP name and SSID as parameters to the DHCP to assign the IP address based on the AP and SSID combination.

Cisco WLC allows you to send different parameters as part of Option 82, such as AP-LOCATION, AP-ETH-MAC, and AP-MAC: SSID, among many others.

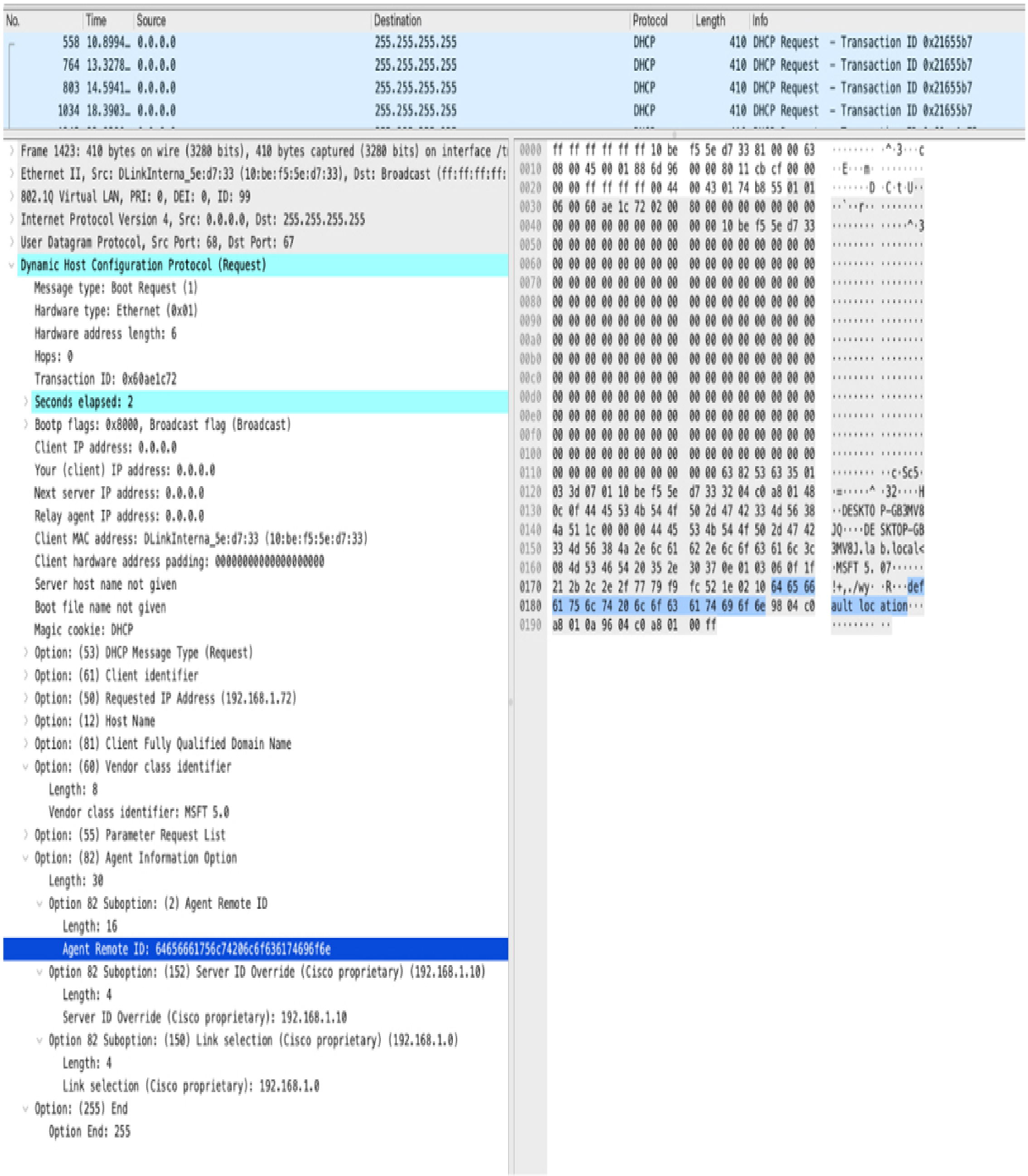
Example 5-1 show a CLI configuration for adding Option 82 details on Cisco Catalyst 9800 WLC.

Example 5-1 *Option 82 Configuration on Cisco Wireless LAN Controller*

```
CiscoWLC9800(config)#wireless profile policy DHCPOP82
CiscoWLC9800(config-wireless-policy)#vlan 100
CiscoWLC9800(config-wireless-policy)#ipv4 dhcp opt82
```

```
CiscoWLC9800(config-wireless-policy)#ipv4 dhcp opt82 ascii
CiscoWLC9800(config-wireless-policy)#ipv4 dhcp opt82 rid
CiscoWLC9800(config-wireless-policy)#ipv4 dhcp opt82 format ?
  ap_ethmac     Enable dhcp AP_EthMac
  ap_location   Enable AP_Location
  apmac         Enable ApMac
  apname        Enable APNAME
  policy_tag    Enable Policy_tag
  ssid          Enable SSID (Delimiter is ':' when used along with
  vlan_id       Enable VLAN_ID (Delimiter is ':' when used along w
options)
```

Figure 5-4 shows the DHCP Option 82 with AP location as remote ID sent by the Cisco WLC to the DHCP server. The hex value in the packet dump translated to the “Default Location” keyword.



```
No.    Time       Source    Destination       Protocol  Length  Info
  558  10.8994…   0.0.0.0   255.255.255.255   DHCP         410  DHCP Request  - Transaction ID 0x21655b7
  764  13.3278…   0.0.0.0   255.255.255.255   DHCP         410  DHCP Request  - Transaction ID 0x21655b7
  803  14.5941…   0.0.0.0   255.255.255.255   DHCP         410  DHCP Request  - Transaction ID 0x21655b7
 1034  18.3903…   0.0.0.0   255.255.255.255   DHCP         410  DHCP Request  - Transaction ID 0x21655b7
```

```
> Frame 1423: 410 bytes on wire (3280 bits), 410 bytes captured (3280 bits) on interface /t
> Ethernet II, Src: DLinkInterna_5e:d7:33 (10:be:f5:5e:d7:33), Dst: Broadcast (ff:ff:ff:ff:
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 99
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
v Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x60ae1c72
  > Seconds elapsed: 2
  > Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: DLinkInterna_5e:d7:33 (10:be:f5:5e:d7:33)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Request)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.1.72)
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  v Option: (60) Vendor class identifier
        Length: 8
        Vendor class identifier: MSFT 5.0
  > Option: (55) Parameter Request List
  v Option: (82) Agent Information Option
        Length: 30
      v Option 82 Suboption: (2) Agent Remote ID
            Length: 16
            Agent Remote ID: 64656661756c74206c6f636174696f6e
      v Option 82 Suboption: (152) Server ID Override (Cisco proprietary) (192.168.1.10)
            Length: 4
            Server ID Override (Cisco proprietary): 192.168.1.10
      v Option 82 Suboption: (150) Link selection (Cisco proprietary) (192.168.1.0)
            Length: 4
            Link selection (Cisco proprietary): 192.168.1.0
  v Option: (255) End
        Option End: 255
```

```
0000  ff ff ff ff ff ff 10 be  f5 5e d7 33 81 00 00 63   ········ ·^·3···c
0010  08 00 45 00 01 88 6d 96  00 00 80 11 cb cf 00 00   ··E···m· ········
0020  00 00 ff ff ff ff 00 44  00 43 01 74 b8 55 01 01   ·······D ·C·t·U··
0030  06 00 60 ae 1c 72 02 00  80 00 00 00 00 00 00 00   ··`··r·· ········
0040  00 00 00 00 00 00 00 00  00 00 10 be f5 5e d7 33   ········ ·····^·3
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0110  00 00 00 00 00 00 00 00  00 00 63 82 53 63 35 01   ········ ··c·Sc5·
0120  03 3d 07 01 10 be f5 5e  d7 33 32 04 c0 a8 01 48   ·=·····^ ·32····H
0130  0c 0f 44 45 53 4b 54 4f  50 2d 47 42 33 4d 56 38   ··DESKTO P-GB3MV8
0140  4a 51 1c 00 00 00 44 45  53 4b 54 4f 50 2d 47 42   JQ····DE SKTOP-GB
0150  33 4d 56 38 4a 2e 6c 61  62 2e 6c 6f 63 61 6c 3c   3MV8J.la b.local<
0160  08 4d 53 46 54 20 35 2e  30 37 0e 01 03 06 0f 1f   ·MSFT 5. 07······
0170  21 2b 2c 2e 2f 77 79 f9  fc 52 1e 02 10 64 65 66   !+,./wy· ·R···def
0180  61 75 6c 74 20 6c 6f 63  61 74 69 6f 6e 98 04 c0   ault loc ation···
0190  a8 01 0a 96 04 c0 a8 01  00 ff                     ········ ··
```

Figure 5-4 *DHCP Option 82 Added by Cisco WLC*

Zero Trust Approach to Dynamic Addressing

In the context of a zero trust security model, DHCP can be configured with security features like DHCP snooping, IP Source Guard (IPSG), port security, and Dynamic ARP Inspection (DAI), which ensure that only authorized devices receive network access. By validating each device's identity and the integrity of DHCP communications, these measures help enforce the principle of least privilege, a core intent of zero trust, reducing the risk of unauthorized access and network-based attacks. To understand this better, let’s look at common DHCP attacks and ways to mitigate them. You will notice that many features associated with DHCP help with the overall security of the network.

Rogue DHCP Servers

In a rogue type of attack, any attacker can introduce a rogue DHCP server onto the network to provide the incorrect IP address, DNS server, gateway, or other configuration information such as a TFTP server to fetch boot files. Although the IP address allotted by the rogue DHCP server may result in network outage due to address conflicts, other configuration parameters like DNS and gateway may redirect the client to websites hosting malware or could result in man-in-the-middle (MITM) attacks.

DHCP snooping is the common solution to avoid such rogue DHCP server issues. This feature provides network security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding database, also referred to as a DHCP snooping binding table. It acts like a firewall between untrusted hosts and DHCP servers.

In Figure 5-5, two DHCP servers are connected to switches. The active DHCP server is on the left switch, and the standby DHCP server is on the right switch (as assumed). At the bottom right, you see a legitimate device trying to get an IP address. Now, imagine if a hacker named Sam runs DHCP server software on his computer. Who do you think will respond first to the DHCP discover message? Will it be the legitimate DHCP server or hacker Sam’s DHCP server software?

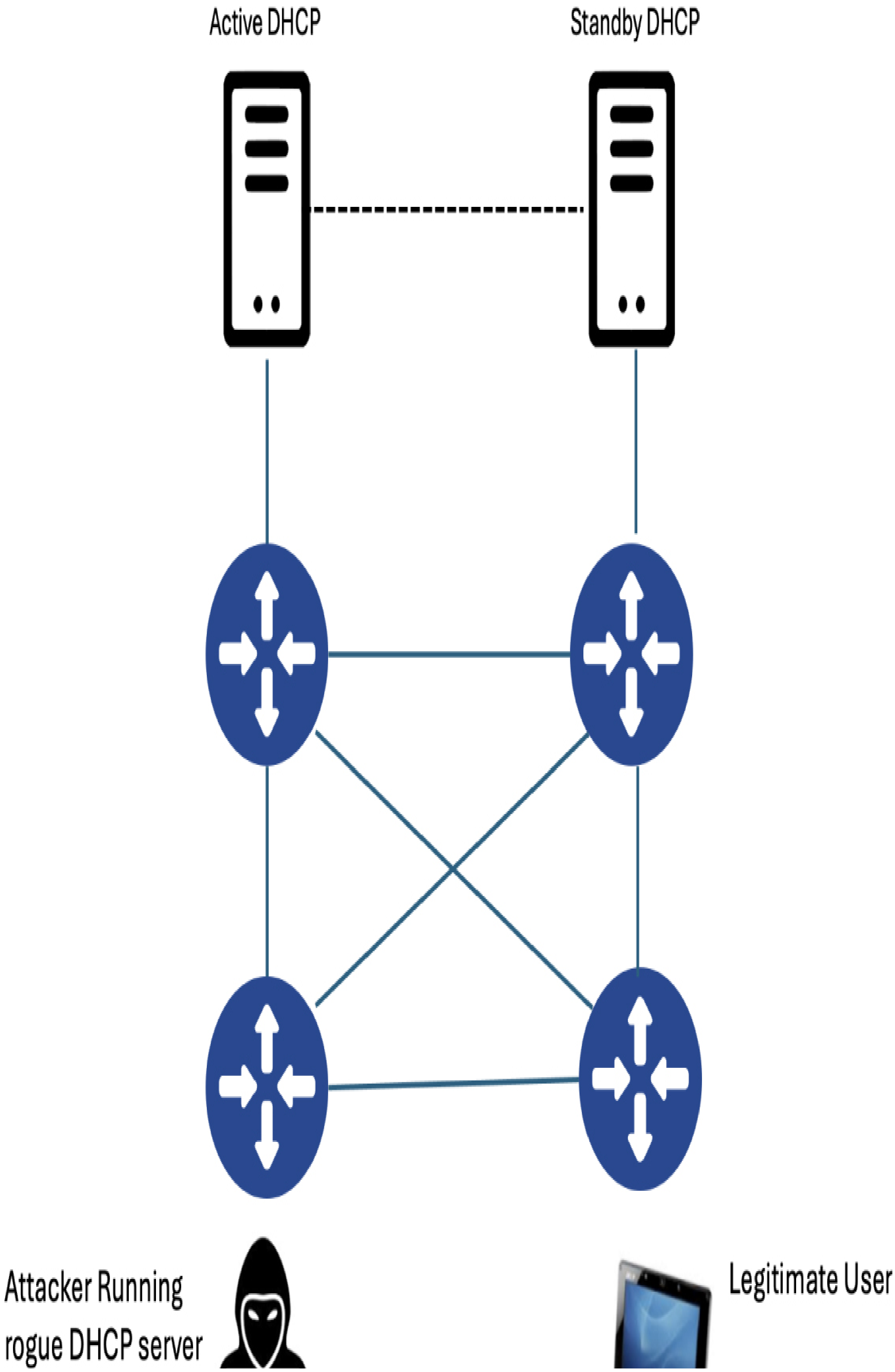

Active DHCP
Standby DHCP
Attacker Running rogue DHCP server
Legitimate User

Figure 5-5 *Rogue DHCP Attack*

In larger networks, a central DHCP server is usually found in the server farm. If an attacker sets up a DHCP server in the same subnet, they might respond faster to a client’s DHCP discover message. If they succeed, the attacker could assign their IP address as the client’s default gateway, enabling a man-in-the-middle attack. This attacker might also set their IP address as the DNS server to spoof websites. The attacker could also flood the DHCP server with discover messages to exhaust its IP address pool. To stop this, you can use DHCP snooping! This feature lets you configure your switches to monitor DHCP discover and offer messages.

Interfaces that connect to clients should never be allowed to send a DHCP offer message. You can enforce this by making them untrusted. An untrusted interface will block DHCP offer messages. Only an interface configured as trusted is allowed to forward DHCP offer messages. You can also rate-limit interfaces so they can’t send an unlimited number of DHCP discover messages. This will prevent attacks from depleting the DHCP pool.

The DHCP snooping table shown in Example 5-2 is different from the DHCP binding table and has additional information such as VLAN and interface information. This information is used by many other security features such as IP Source Guard.

Example 5-2 *Output from the* ***show ip dhcp*** *Command*

```
SW1#show ip dhcp snooping


<--output ommitted -->


Interface                      Trusted      Rate limit (pps)
------------------------       -------      ----------------
FastEthernet0/1                no           10
```

```
FastEthernet0/2              yes        unlimited


SW1#show ip dhcp snooping binding
MacAddress          IpAddress        Lease(sec)  Type           V
------------------  ---------------  ----------  -------------  -
00:0C:29:28:5A:6B   192.168.1.1      85632       dhcp-snooping
```

DHCP rogues could also be introduced in unprotected Wi-Fi networks such as public hotspots with no Layer 2 encryption. In a typical Wi-Fi hotspot, users connect to open SSID at Layer 2 and later perform the Layer 3 web auth. In such a case, an attacker can introduce a rogue DHCP, providing incorrect DNS details with IP assignment from the correct pool. So, when the client is authorized, it uses the incorrect DNS, and the user may be redirected to fake websites, resulting in potential financial loss and privacy breaches.

In the Wi-Fi scenario in addition to DHCP snooping, which is typically implemented at the wireless controller level, you can also enable peer-to-peer blocking. This stops any direct communication between Wi-Fi clients and allows all clients to access the Internet/upstream data only. The DHCP Required option could also be enabled for SSID, which forces a client to have the IP assigned by a configured DHCP before it can pass traffic on an SSID. This Cisco-specific feature is available on most wireless LAN controllers. At the time of writing this chapter, the DHCP required configuration was not supported on Cisco SD-Access SSID configuration. You can also use advanced features like Opportunistic Wireless Encryption (OWE) defined as part of IETF RFC 8110 with guest SSID. The only problem is that because these are new standards, many of the legacy clients still don’t support them.

Note

In a Cisco Unified Wireless architecture with wireless LAN controller and access points, only Cisco WLC acts as a DHCP relay agent, and it can add additional details via Option 82. This means that an SSID needs to be centrally switched to make use of this feature. For SSIDs that are configured for local switching of traffic at AP, DHCP requests are directly sent on the access switch. In this case, configuration on the client VLAN will be applied. Cisco Access Point in Flex Mode does not perform any DHCP snooping function, but it does basic address learning and client profiling.

DHCP Starvation

DHCP starvation is a kind of denial-of-service (DoS) attack. In this type of attack, an attacker floods the DHCP server with bogus DISCOVER packets until the DHCP server exhausts its supply of the IP addresses. Once that happens, legitimate clients will not receive the IP address. In another scenario, the attacker can introduce a rogue DHCP resulting in a man-in-the-middle attack. The variation of this attack can use packets like DHCP Inform to flood the DHCP server.

You can protect your network from DHCP starvation attacks in the following ways:

1. **Enabling Port Security:** You can use port security with dynamic and static MAC addresses to restrict ingress traffic by limiting allowed MAC addresses. When you assign secure MAC addresses, the port won't forward traffic from sources outside the defined addresses. Limiting secure MAC addresses to one gives the attached device full bandwidth port.

2. **Configuring 802.1x:** You can enable 802.1x switch port authentication on your network. This approach allows only authorized clients to be connected to your network. Wi-Fi with WPA1/2/3 enterprise mode will use 802.1x by default for the authorization of all clients on the network. For any legacy clients not supporting 802.1x, you can configure the MAC-based port security on the switch port. Although it is easy to bypass MAC-based security with MAC address spoofing, it may help with the DHCP starvation issue because it will allow only a specific number of addresses to communicate from that port. When an attacker tries to send DHCPDISCOVER using different MAC addresses, the port will shut down.

3. **Configuring Rate Limit:** You can implement rate limiting on DHCP requests per port. This approach limits the number of DHCP requests that can be sent from a single port, mitigating the effect of an attacker flooding the network.
4. **Enabling DHCP Snooping:** As discussed earlier in this chapter, this feature allows switches to differentiate between trusted and untrusted DHCP messages, and only allows legitimate DHCP requests from trusted ports.

DHCP Man in the Middle

DHCP man-in-the-middle (MITM) attacks occur when an attacker intercepts the DHCP communication between a client and the legitimate DHCP server. By doing so, the attacker can manipulate the IP configuration provided to the client, redirect traffic, or gain unauthorized access to sensitive information. You can use protection mechanisms such as DHCP snooping and port security. In addition, you can also use Dynamic ARP Inspection (DAI).

Dynamic ARP Inspection is a security feature that validates ARP packets in a network. It intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. This capability protects the network from certain man-in-the-middle attacks. Dynamic ARP Inspection determines the validity of an ARP packet based on valid IP-to-MAC address bindings stored in the DHCP snooping binding database. DAI intercepts ARP packets on untrusted ports and checks the IP-MAC binding against the DHCP snooping table. If the binding is not valid, the packet is dropped. This prevents attackers from using spoofed ARP messages to perform MITM attacks, enhancing the overall security of the DHCP process.

DHCP Options

Dynamic Host Configuration Protocol options are settings that can be sent to devices on a network when they request an IP address. These options include details like the network's default gateway, DNS server, or specific configurations that devices need to connect properly.

Some DHCP options can enhance security by specifying which DNS servers or NTP servers should be used, reducing the risk of devices connecting to malicious or unauthorized servers. In a zero trust approach, DHCP options can be used to enforce security policies, ensuring that devices receive only the necessary configurations and are monitored for compliance with the network's security standards.

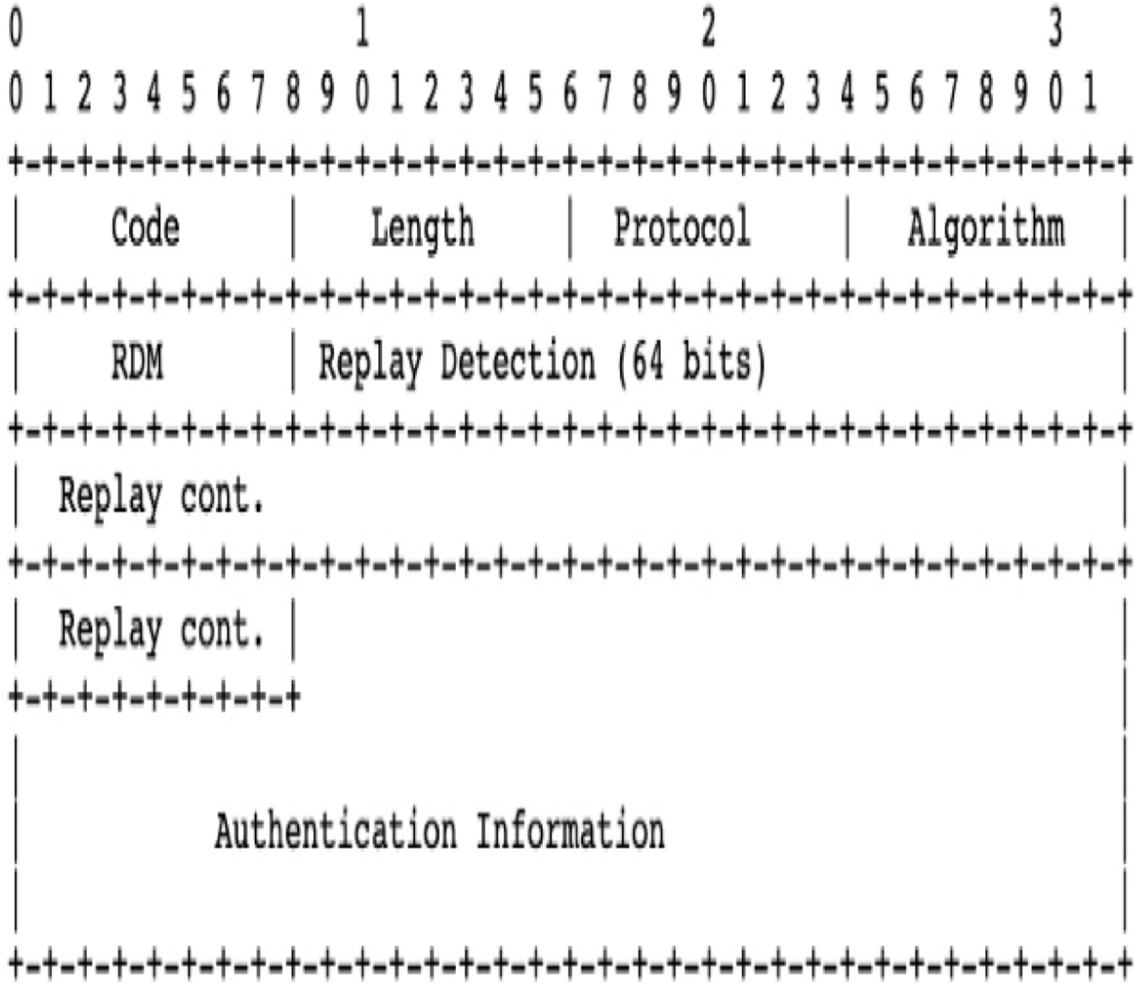
Table 5-1 shows some of the common DHCP options.

Table 5.1 *Common DHCP Options and Purpose*

DHCP Option	Purpose	Use Case
1	Assign subnet mask	Enabling client Network Connectivity by providing subnet mask
3	Assign default router	Enabling client Network Connectivity by providing Gateway address
6	Assign DNS	Enabling client to Resolve names by providing DNS server details
12	Hostname of the client	Allows assigning names to devices like IoT devices
15	Domain name assignment	Specifies the domain name that client should use as suffix when resolving hostnames via the Domain Name System.
42	List of NTP servers	Allows Time syncing across a network
43	Vendor-specific custom configuration	Cisco uses this to prime the wireless AP for a specific wireless LAN controller
82	Circuit-specific information	Assigning IP addresses based on additional parameters such as AP location, SSID, AP-Name, and MAC address

DHCP Authentication

In the preceding sections, you learned about different attacks that can be launched in the DHCP process. Some attacks target DHCP servers like starvation attacks, whereas others impact the client by adding rogue DHCP servers in the network. To solve some of these issues, the authenticated DHCP process is proposed as part of RFC 3118. It is a method to secure the process of assigning IP addresses in a network. The main idea is to ensure that only trusted devices can request and trusted DHCP servers can assign IP addresses. This is done by adding a special authentication option (Option 90) in DHCP messages. Figure 5-6 shows the format for DHCP Option 90.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |    Length     |  Protocol     |   Algorithm   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     RDM       | Replay Detection (64 bits)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Replay cont.                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Replay cont. |                                               |
+-+-+-+-+-+-+-+-+                                               |
|                                                               |
|           Authentication Information                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5-6 *DHCP Option 90 Format*

You can decode the fields of Figure 5-6 as follows:

- **Code value:** 90

- **Length:** Length of the entire option
- **Protocol:** The authentication method used
- **Algorithm:** The specific algorithm used within protocol field
- **Replay Detection Method (RDM):** The type of replay detection

RFC 3118 defines two types of authentication methods:

- **Protocol Authentication:** In this method, the protocol field is set to 0. This method uses a keyed message authentication code (MAC) algorithm to ensure the message's integrity and authenticity. The MAC is calculated based on the content of the DHCP message and a shared secret key known as a configuration token, which is carried in the Authentication Information fields of Option 90. This hash calculation excludes the value of additional relay agent options like Option 82. The configuration token can be used to pass a plain-text configuration token, but it only offers weak authentication of the server's identity and doesn't secure the actual message. This method is effective only for basic protection against accidentally setting up unauthorized DHCP servers.
- **Delayed Authentication:** In this method, the protocol field is set to 1. In delayed authentication, the client asks for authentication in its DHCPDISCOVER message, and the server responds with a DHCPOFFER message that includes authentication details. These details include a nonce value generated by the server, which is used as a message authentication code to ensure both the message and the server's identity are verified.

DHCP Option 90 is less commonly implemented compared to other DHCP options, but it's important for environments where enhanced security is needed, such as financial institutes, hospitals, or any place where the possibility of someone attaching rogue DHCP is high.

IPv6 Address Assignment

As the Internet grows and more devices come online, the need for a new system of IP addresses becomes essential. The old IPv4 system, which was created in the early days of the Internet, provided about 4.3 billion unique addresses. Although that number seemed huge at the time, it’s not enough for the billions of devices now connecting to the Internet—from smartphones, computers, smart home gadgets, and even cars. This is where IPv6 comes into play. IPv6 is the latest version of the Internet Protocol; it is 128 bits long and supports a vast number of addresses (340 undecillion addresses). While the primary reason for IPv6 was to create a new address range, designers of IPv6 also took this opportunity to remove some shortcomings of IPv4. IPv6 offers the following additional benefits compared to IPv4:

- **Stateless Autoconfiguration**—This feature allows clients to calculate IPv6 addresses without a DHCP server.
- **Use of Extension Headers**—These headers provide additional information and options for packet processing such as routing, fragmentation, and security without increasing the size of the header.
- **Routers Exempted from Fragmentation**—IPv6 puts the responsibility of handling fragmentation to sending clients; this reduces the processing load on the routers, resulting in more efficient packet forwarding and improved overall network performance.
- **Less Dependency on NAT/PAT**—IPv6 provides vast address space, allowing every device to have a unique global address. This eliminates the need for NAT/PAT. However, in specific cases, you might have to use NAT flavors like NAT64 that translate IPv6 addresses to IPv4, allowing IPv6-only devices to communicate with IPv4-only servers. You will find these requirements in case of legacy applications that do not support dual-stack or IPv6 addressing.

IPv4 had limited ways to assign addresses to a client, primarily DHCP and Static. IPv4 and IPv6 support unicast, multicast, and anycast address types. But IPv6 is different. It has new types of addresses that make data delivery better and more efficient and allows client communication without the use of DHCP servers. Knowing these new address types is important for getting the most out of IPv6 and making networks work better. Figure 5-7 shows the different address types used in IPv6.

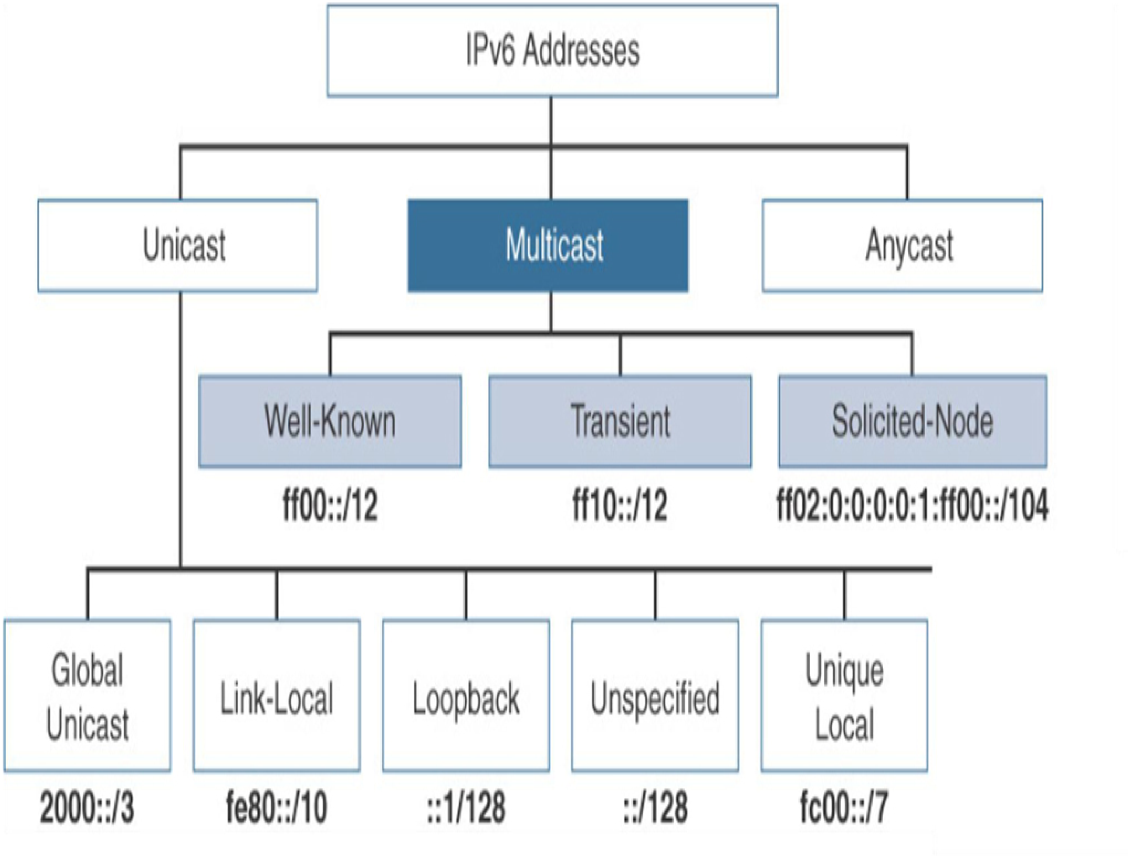


Figure 5-7 *IPv6 Address Types*

IPv6 addresses can be divided into three major categories: unicast, multicast, and anycast. Unicast address space allows unique identification of an endpoint and can be divided further into the following categories:

- **Global Unicast Address (GUA):** These are like IPv4 public addresses. These addresses have the prefix 2000::/3 (addresses starting with binary 001) and are Internet routable.
- **Link-Local Addresses:** These addresses are not routable and are used on the same network only. When you enable IPv6 on the interface, this address is automatically calculated by the device. These addresses play a crucial role in the IPv6 neighbor discovery process. IPv6 does not use ARP for finding the Layer 2 address of a

host; instead, it uses the neighbor discovery process. These addresses start with fe80::/10.

- **Unique Local Address (ULA):** These addresses are like IPv4 private address ranges. These addresses are not Internet-routable, and you will need to use IPv6 NAT66. As you know, the concept of IPv6 was to provision each endpoint with a public routable IP address and move away from NAT. So you should not consider a Unique Local Address[nd]based IPv6 design for your network. Instead, you can use ULA along with GUA for a client; this provides flexibility in designing and securing the routing for each scope. ULA can be used within or between site communication and GUA for Internet traffic.

Note

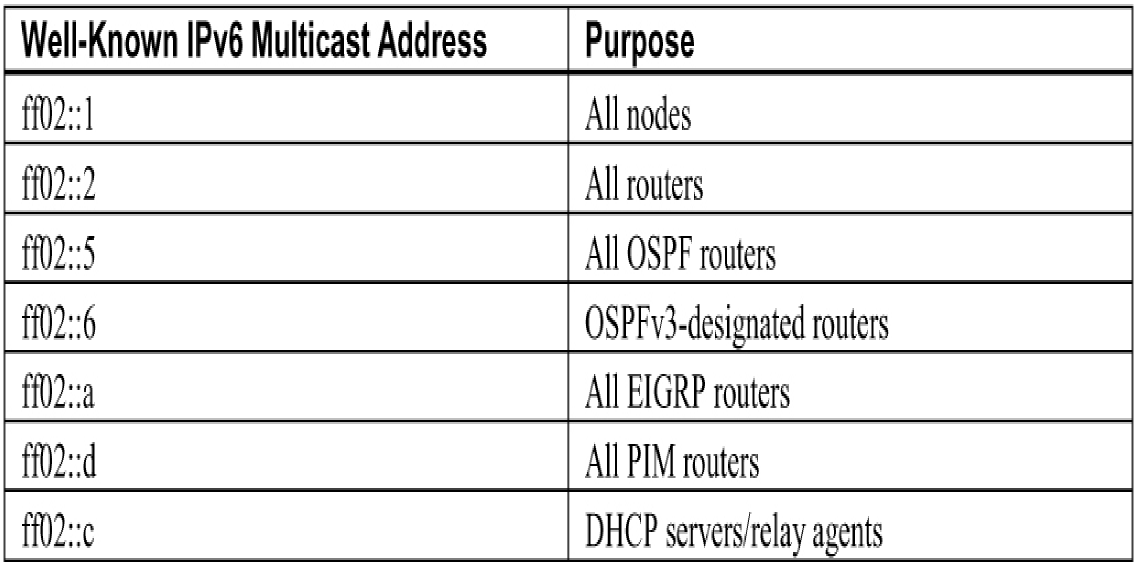
IPv6 network design is out of the scope of this book. It is recommended that you adopt an IPv6 address strategy based on the business needs of your organization. You also need to devise a transition plan that typically includes first adopting a dual stack with the end state of IPv6-only network in case of brownfield deployments.

IPv6 multicast addresses are special addresses used to send packets to multiple destinations simultaneously. They are defined by specific address prefixes and can be categorized into different types based on their scope and purpose. IPv6 does not offer the concept of broadcast, so multicast addresses play a critical role in IPv6 communication. Next, let’s look at the main types of IPv6 multicast addresses.

Well-Known Multicast

IPV6 well-known multicast addresses are similar to IPv4 well-known multicast address space. These addresses are predefined and reserved for special purposes. These addresses start with the prefix ff00::/12. The first three hexadecimal digits of an address are always ff0. Table 5-2 shows some of the well-known IPv6 multicast addresses.

Table 5.2 *Well-Known IPv6 Multicast Addresses*



Well-Known IPv6 Multicast Address	Purpose
ff02::1	All nodes
ff02::2	All routers
ff02::5	All OSPF routers
ff02::6	OSPFv3-designated routers
ff02::a	All EIGRP routers
ff02::d	All PIM routers
ff02::c	DHCP servers/relay agents

Transient Multicast IPv6 Addresses

Transient multicast IPv6 addresses are temporary and are used for specific multicast groups that are dynamically assigned or created—for example, streaming media and online gaming, where addresses are required for a specific session duration.

Neighbor Discovery in IPv6

IPv6 Neighbor Discovery (ND) is a protocol used to discover other devices on the same local network, determine their link-layer addresses, and maintain reachability information. It replaces the IPv4 Address Resolution Protocol (ARP) and has several key functions. It uses the following four packet exchanges for this purpose:

- **Neighbor Solicitation (NS):** A device sends a Neighbor Solicitation message as a multicast to all devices on the local network. This message includes the IPv6 address of the device the sender wants to find.
- **Neighbor Advertisement (NA):** The device with the target IPv6 address replies with a Neighbor Advertisement message. This message includes the link-layer address of the device that was queried.

- **Router Solicitation (RS):** A device sends a Router Solicitation message to the all-routers multicast address to ask routers to send Router Advertisements.
- **Router Advertisement (RA):** Routers respond to Router Solicitations with Router Advertisement messages. These messages include network prefixes, default gateway information, and other network configuration details. Routers also send RA periodically without clients sending RS.

Solicited-Node Multicast Addresses

Solicited-node multicast addresses in IPv6 play a crucial role in the Neighbor Discovery Protocol (NDP), particularly in the process of determining the link-layer address of a neighbor (similar to ARP in IPv4). These addresses are specifically designed to minimize the scope of Neighbor Solicitation messages, reducing unnecessary traffic on the network. When an interface is configured with an IPv6 unicast address, a solicited-node multicast address is generated automatically based on the unicast address for this interface, and the node joins the multicast group. This means every node will create a solicited multicast address for itself. Example 5-3 shows the output of the show IPv6 interface on a Cisco 9300 catalyst switch. You will notice that the switch has created and joined the FF02::1:FFA2:C5D solicited-node address.

Example 5-3 *IPv6 Addresses on Switch VLAN Interface*

```
Cisco-C9300-1#show ipv6 interface vlan 99
Vlan99 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::4A2E:72FF:FEA2:C5D
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8:0:2::1, subnet is 2001:DB8:0:2::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:2
    FF02::1:FF00:1
    FF02::1:FFA2:C5D
    FF05::1:3
```

When an IPv6 node wants to discover the link-layer address of another node (that is, it wants to know the MAC address associated with an IPv6 address), it sends a Neighbor Solicitation message to the Solicited-Node multicast address derived from the target IPv6 address. Only the node with the corresponding IPv6 address will recognize its Solicited-Node multicast address and respond with a Neighbor Advertisement message containing its link-layer address.

Anycast Addresses

An anycast address in IPv6 is a kind of address you can give to many devices, usually in different places. When you send data to an anycast address, it goes to the closest or best device based on the network's routing rules. The main idea of using anycast addresses is to make sure services work quickly by sending requests to the nearest server or device. The fundamental concept of the anycast address is the same in both IPv4 and IPv6. A typical use of anycast is for service resilience. By deploying service using the anycast address, organizations can ensure higher availability and resilience. When a node fails, traffic is automatically redirected to another without making any changes to the client configuration.

Address Assignment in IPv6

You can always assign static IPv6 addresses to the devices. Static assignment of addresses is cumbersome, and IPv6 has many autoconfiguration mechanisms that allow a client to calculate its addresses. Dynamic IPv6 address assignment can be managed centrally using DHCPv6 servers, or you can let the client calculate its address using the Stateless Address Autoconfiguration mode. Figure 5-8 shows the address assignment modes in IPv6.

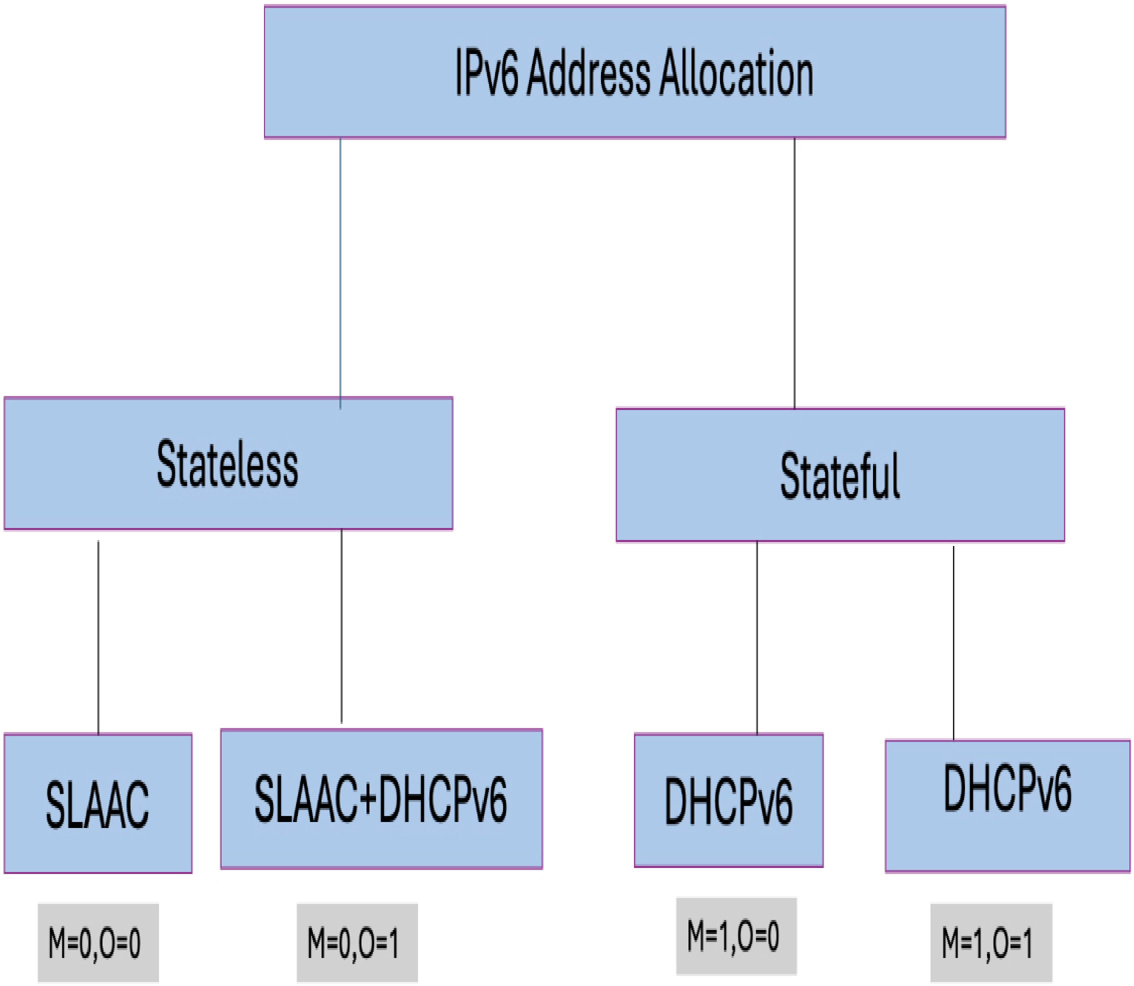


Figure 5-8 *IPv6 Address Assignment Methods*

The M (Managed) and O (Other) flags are part of the IPv6 Neighbor Discovery Protocol (NDP) and are used in Router Advertisement (RA) messages to inform IPv6 hosts about how they should obtain their network configuration information, such as IP addresses and other configuration details like DNS servers.

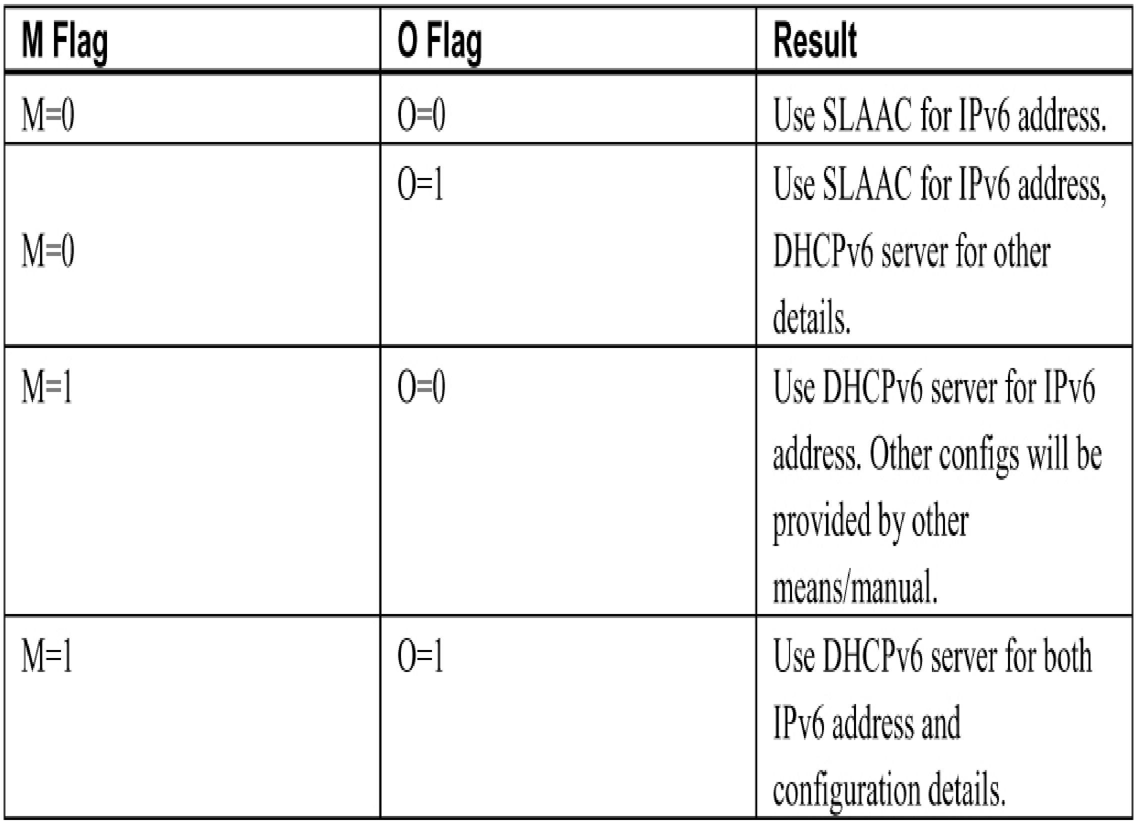
- **Stateless Address Autoconfiguration (SLAAC):** This IPv6 method allows a device to automatically configure its IP address and other related settings without the need for a DHCP server. SLAAC is a key feature in IPv6, designed to simplify the process of address assignment and network configuration. You will notice that both M and O flags are set to 0 in pure SLAAC mode. This means that the client needs to calculate the address itself based on the prefix shared by a router in the RA message. RA messages are periodic messages sent by the router announcing its prefix to the network. RA messages could also be sent in response to requests from an IPv6 client. An RA message also sets flag A= 1, which tells the client that it can use the prefix advertised in the RA message for address calculation. It is important to note that with SLAAC, the client gets only the basic information like IPv6 address and gateway, but other details such as DNS, TFTP, or custom information cannot be provided to clients. This method is suitable for simple networks where basic IPv6 client connectivity is required within the local network. While SLAAC makes the configuration simpler, it is difficult to have central control of address tracking as clients compute their IPv6 addresses using schemes like EUI-64.

- **SLAAC +DHCPv6:** The other variation of stateless configuration uses the RA message from the router to calculate the client address and default gateways, but explicitly tells the client to reach out to a configured DHCPv6 server to get additional parameters like DNS or TFTP. In this option, the M flag is set to 0, which tells the client to use SLAAC for IPv6 address calculation, but the O flag is set to 1, directing the client to connect to the DHCPv6 server to get other configuration details.

- **Stateful DHCP:** This option explicitly tells the client to use the DHCPv6 server to get its address. Again, the M and O flags direct the clients if they need to use DHCPv6 for other configurations: M =1, O = 0. This combination is less common and is useful only in scenarios where your clients obtain only an IPv6 address from the server and not the other configurations. In the most commonly used method, both M and O flags are set to 1, and the client obtains the IP address and other details from the DHCP server.

Table 5-3 summarizes the M and O flag options and their impact on client addressing.

Table 5.3 *M and O Flag Impact on Client IPv6 Address*



M Flag	O Flag	Result
M=0	O=0	Use SLAAC for IPv6 address.
M=0	O=1	Use SLAAC for IPv6 address, DHCPv6 server for other details.
M=1	O=0	Use DHCPv6 server for IPv6 address. Other configs will be provided by other means/manual.
M=1	O=1	Use DHCPv6 server for both IPv6 address and configuration details.

We hope you have clarity about the DHCPv6 address assignment methods. However, did you notice we mentioned that the client will get the IPv6 address from the DHCPv6 server in the stateful address assignment method and explicitly left the details about the default gateway? The reason is that, in DHCPv6 address assignment, the default gateway still needs to be calculated based on the RA messages similar to the SLAAC. The DHCPv6 server will provide you only the IPv6 address and other configuration details. Confused? Let’s look at this entire process in detail with the help of Figure 5-9.

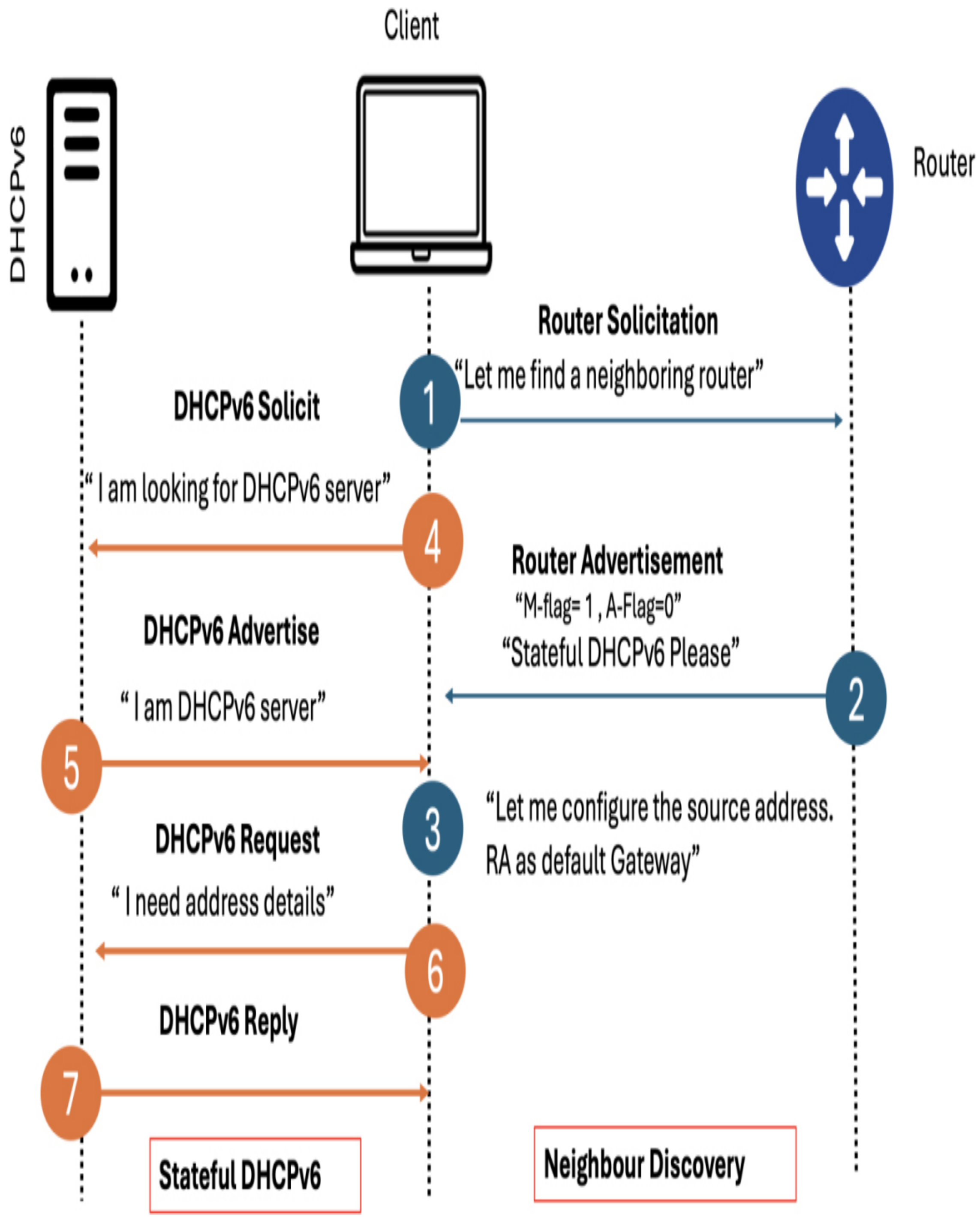


Figure 5-9 *IPv6 Address Assignment Using DHCPv6 Server*

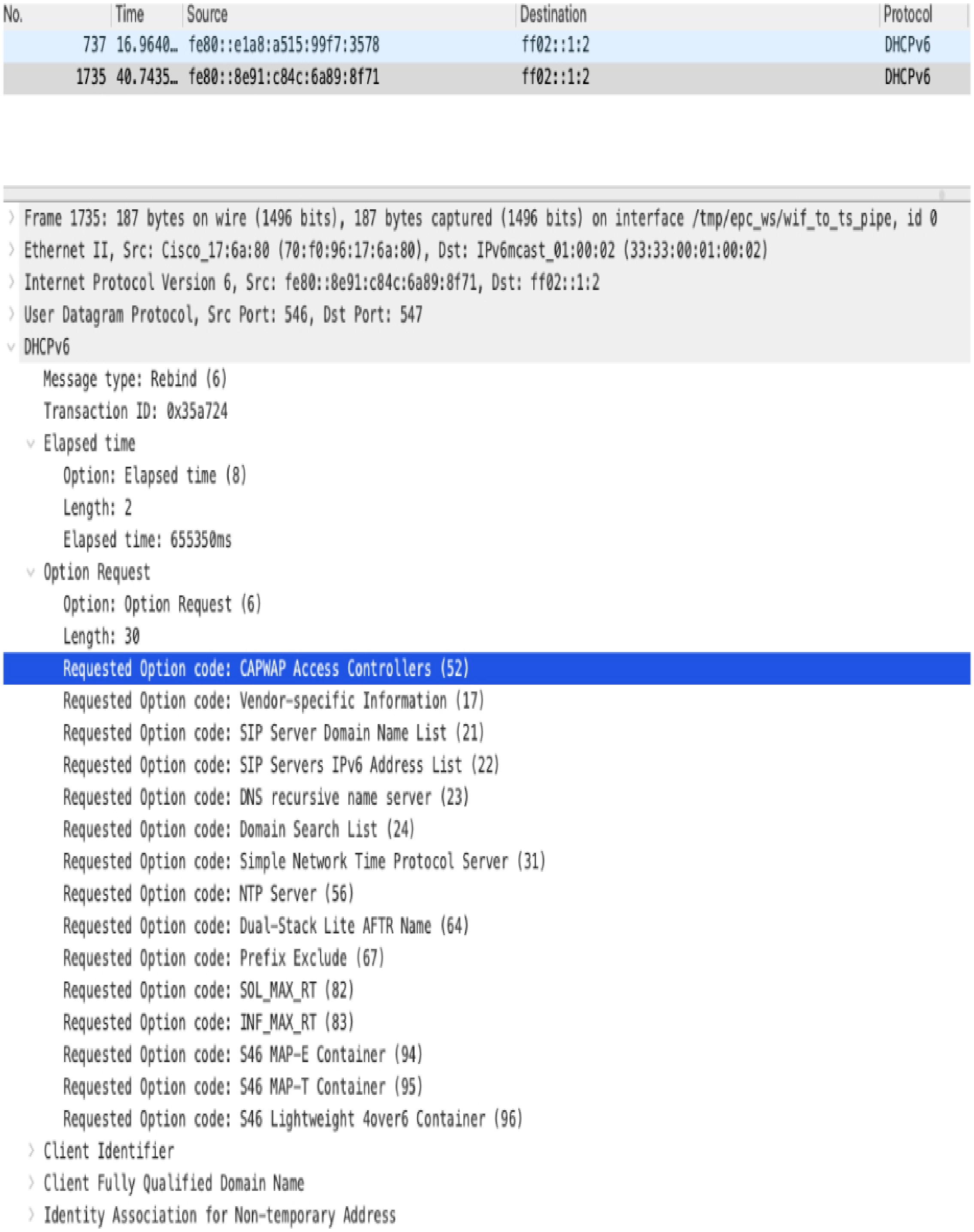
The stateful DHCPv6 process is a combination of Neighbor Discovery and DHCPv6 communication. You will explicitly configure the router or VLAN interface with M flags set to 1 and A flags set to 0.

1. After the client boots up, it sends a Router Solicitation message on the local network. This message is for finding the local router on the network.
2. After receiving this message, a router replies with a Router Advertisement message with an A flag set to 0, an M flag set to 1, and an O flag set to 1 (assuming you want the client to obtain other configurations from the DHCPv6 server as well).
3. When the client processes this RA message and looks at the various flags, it understands that it cannot use the prefix sent in RA for IPv6 address calculation, and it now needs to find a DHCPv6 server. But at this point, it uses the source address of RA as the default gateway for itself.
4. The client then sends the DHCPv6 solicit message to find the DHCPv6 server on the local network.
5. The available DHCPv6 server (or DHCPv6 relay) will respond with a DHCPv6 advertise message.
6. The client then sends the DHCPv6 request message to the DHCPv6 server.
7. The DHCPv6 server sends the reply message with the assigned address.

After completing these steps, the client is now ready to communicate with the external world. You might notice that the client might have multiple IPv6 addresses assigned to it. From a network design perspective, it means that now TCAM (Ternary content-addressable memory) on the routers and switches needs to store additional information per client; therefore, you need to make sure the routers have enough TCAM before you transition to dual-stack or IPv6-only network architecture.

DHCPv6 Options

DHCPv6 options work similarly to DHCPv4 options and allow the carrying of additional information for the clients. However, it is important to note that different option numbers might be used to carry the same information in IPv6 when compared to IPv4. As an example, Option 43 is used in IPv4 to carry details of the Cisco wireless LAN controller to the access points while Option 52 is used in IPv6 to provide the same information to the access points. Figure 5-10 shows the DHCPv6 rebind message from the Cisco wireless access point (AP) with Option 52 included. The Rebind message is used by a DHCPv6 client when it has been unable to contact its original DHCPv6 server to renew its lease. This situation typically arises when the client's lease is approaching expiration, and it hasn't received a response to its Renew message, which is the first step a client takes to renew its lease. In this case, we had to delete the DHCPv6 server configuration on the switch to capture this packet from AP.



```
No.     Time       Source                        Destination    Protocol
    737 16.9640…  fe80::e1a8:a515:99f7:3578     ff02::1:2      DHCPv6
   1735 40.7435…  fe80::8e91:c84c:6a89:8f71     ff02::1:2      DHCPv6
```

```
> Frame 1735: 187 bytes on wire (1496 bits), 187 bytes captured (1496 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0
> Ethernet II, Src: Cisco_17:6a:80 (70:f0:96:17:6a:80), Dst: IPv6mcast_01:00:02 (33:33:00:01:00:02)
> Internet Protocol Version 6, Src: fe80::8e91:c84c:6a89:8f71, Dst: ff02::1:2
> User Datagram Protocol, Src Port: 546, Dst Port: 547
v DHCPv6
    Message type: Rebind (6)
    Transaction ID: 0x35a724
  v Elapsed time
      Option: Elapsed time (8)
      Length: 2
      Elapsed time: 655350ms
  v Option Request
      Option: Option Request (6)
      Length: 30
      Requested Option code: CAPWAP Access Controllers (52)
      Requested Option code: Vendor-specific Information (17)
      Requested Option code: SIP Server Domain Name List (21)
      Requested Option code: SIP Servers IPv6 Address List (22)
      Requested Option code: DNS recursive name server (23)
      Requested Option code: Domain Search List (24)
      Requested Option code: Simple Network Time Protocol Server (31)
      Requested Option code: NTP Server (56)
      Requested Option code: Dual-Stack Lite AFTR Name (64)
      Requested Option code: Prefix Exclude (67)
      Requested Option code: SOL_MAX_RT (82)
      Requested Option code: INF_MAX_RT (83)
      Requested Option code: S46 MAP-E Container (94)
      Requested Option code: S46 MAP-T Container (95)
      Requested Option code: S46 Lightweight 4over6 Container (96)
  > Client Identifier
  > Client Fully Qualified Domain Name
  > Identity Association for Non-temporary Address
```

Figure 5-10- *IPv6 DHCPv6 Option 52*

IPv6 First Hop Security

IPv6 First Hop Security is about keeping your network safe right at the start, where devices first connect to the network. It helps to stop bad things like fake addresses, rogue routers, or attackers who want to mess with your network. It checks the IPv6 traffic, making sure only trusted devices and routers are talking, and blocks anything suspicious. This way, it keeps your network more secure from the very beginning. In this section, we look at the common IPv6 first hop security features on Cisco devices.

Rogue RA

In the previous section, you learned about how SLAAC allows clients to compute their IPv6 address based on the prefix value provided by the router in the Router Advertisements (RAs). These RAs also provide the gateway, data link layer address of the router, and miscellaneous options like MTU to the client. But what if an attacker introduces rogue RA messages in the network? The clients will configure themselves with bogus information, which results in attacks like man-in-the-middle and denial-of-service. To avoid these attacks related to RA addresses, you need to ensure that only authorized routers can send RA messages. You can adapt the following solutions to safeguard your network from rogue RAs"

- The easiest approach is to use a feature called RA guard. The Router Advertisement guard feature checks the information in the Router Advertisement message against the settings on the Layer 2 device. If the information matches, the device forwards the message to its destination. If the information doesn't match, the message is blocked. You can configure which parameters you want to validate in the RA guard policy (e.g., managed flag config, prefix list, another config flag). You can even block all the RA packets coming from the host ports.

- You can adopt host isolation approaches to prevent node-to-node communication. This will work only in scenarios where two nodes within the same domain are not supposed to talk to each other. You can use the concept of private VLANS /SGTs (blocking communication with clients using the same SGT) in wired networks or peer-to-peer blocking in Cisco wireless (or commonly known as AP isolation mode). Cisco Meraki also has the concept of port isolation. You will deploy this in guest access scenarios where users are supposed to access the Internet but are not allowed to access other devices on the same network.

- Secure Neighbor Discovery (SEND) is an extension of the Neighbor Discovery Protocol (NDP) in IPv6. It uses cryptographic techniques to ensure that RAs and other NDP messages are authentic and haven't been tampered with. However, SEND is not widely implemented due to its complexity.

DHCPv6 Guard

Rogue DHCPv6 servers can disrupt the network and create security risks. It is important to block unauthorized DHCPv6 on your network. The DHCPv6 Guard feature, when enabled on a switchport, allows trusted servers to send messages and blocks the reply and advertisement messages coming from the unauthorized DHCP servers and relay agents. DHCPv6 Guard is like DHCPv4 snooping but provides granular control to define the DHCPv6 client or server policies. Using these policies, you can filter based on specific prefixes. It is important to note that the DHCPv6 guard does not create any snooping table similar to the DHCP snooping table. Example 5-4 is a sample configuration for Cisco IOS to define DHCP guard policy for the device role server.

Example 5-4 *IPv6 DHCP Guard Sample Configuration*

```
enable
configure terminal
ipv6 access-list acl1
permit host FE80::A8B1:DD01:FE01:F600 any

ipv6 prefix-list dhcpguard permit 2001:0DB7::/64 le 128
ipv6 dhcp guard policy poldhcpv6
   device-role server
   match server access-list acl1
   match reply prefix-list dhcpguard
   preference min 0
   preference max 255
   trusted-port

interface GigabitEthernet 1/0/1
  switchport
  ipv6 dhcp guard attach-policy poldhcpv6
  vlan configuration 1
  ipv6 dhcp guard attach-policy poldhcpv6
```

IPv6 Destination Guard

IPv6 Destination Guard is a security feature found in Cisco switches and routers that helps protect your network from certain types of attacks. The main job of the IPv6 Destination Guard is to make sure that devices on your network are allowed to communicate only with valid IPv6 addresses. This helps prevent attackers from sending fake or harmful traffic to other devices on the network. IPv6 Destination Guard uses a feature called *address gleaning* to learn about all the active destinations on the network. It builds a list of these destinations in a binding table. If a device tries to send traffic to a destination not listed in this table, the traffic is blocked before it can reach that destination.

Before filtering incoming traffic, the device listens to Neighbor Discovery Protocol (NDP) and DHCP messages to learn about the addresses on the network. When a packet arrives and the destination or next hop isn’t already known, the device checks the binding table. If the destination isn’t found in the table, the packet is dropped. If it is found, the device continues with Neighbor Discovery (ND) to complete the process.

Source Guard and Prefix Guard

IPv6 Source Guard and IPv6 Prefix Guard are security features that work at Layer 2 to check where IPv6 traffic is coming from.

- **IPv6 Source Guard:** This feature blocks any data from sources that are not recognized. For example, if the source is not in the binding table or hasn't been learned through Neighbor Discovery (ND) or DHCP, the traffic will be blocked.
- **IPv6 Prefix Guard:** This feature stops devices from sending traffic that doesn't match the allowed or authorized network range. Every device on the network is given an IPv6 address that belongs to a specific range, known as a prefix. IPv6 Prefix Guard makes sure that devices send data using only addresses within this allowed range. If a device tries to send traffic with an address outside of this range, IPv6 Prefix Guard will block it. This feature prevents devices from using unauthorized or incorrect addresses, which could cause network problems or security risks.

These features work together to make sure that only trusted devices can communicate on the network, and that they do so in the correct way.

RA Throttle

As you are aware, Router Advertisement messages are vital in IPv6 communication. These messages are sent by the IPv6 routers to inform the connected devices about the presence of the router, the network prefix, and other information like the default gateway. However, if a router is set to send these messages too often, it could result in unnecessary traffic and congestion on the network. RA throttling works by setting a limit on how often RAs can be sent.

In a Wi-Fi scenario where saving airtime is critical for optimal client performance, these periodic RAs can consume a lot of airtime. To avoid this situation, Cisco wireless LAN controllers and Meraki Cloud dashboard have features to throttle these RAs toward the wireless interface. If an RA is sent in response to the Router Solicitation by the client, it will be allowed by the wireless controller and is sent toward the client. Wireless access points also convert the multicast RA frames to unicast. This saves critical airtime as unicast frames are sent at the client connection rate (which could be up to 1.3 Gbps in some Wi-Fi standards) compared to multicast frames, which are supposed to be sent at the lowest mandatory data rate (which is usually set at 6, 12, or 24 Mbps). Sending frames at lower data rates takes longer and consumes a lot of airtime. Figure 5-11 shows the RA throttling in the Wi-Fi environment.

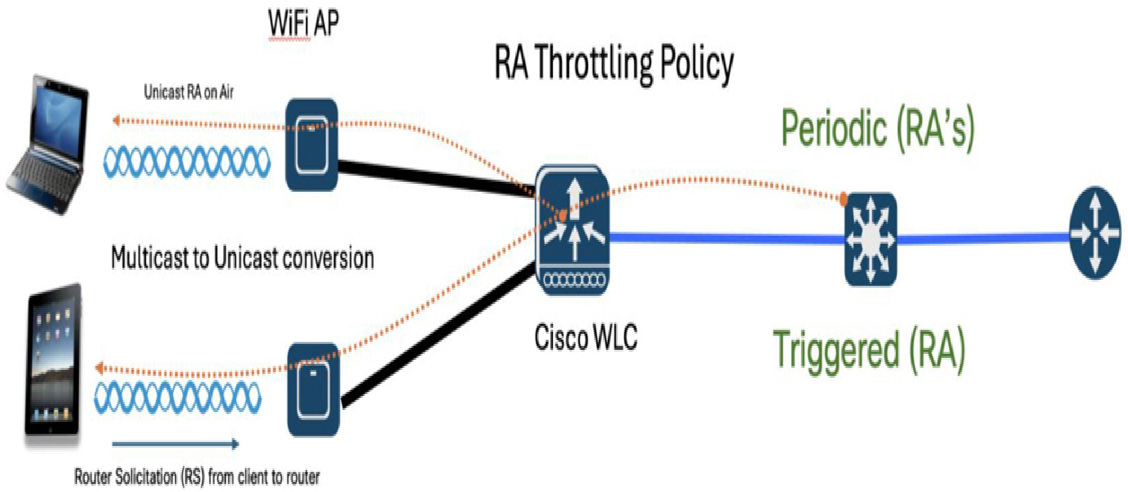


Figure 5-11 *RA Throttling by Cisco WLC*

ND Suppress Multicast

IPv6 Neighbor Discovery (ND) Multicast Suppress is a feature that reduces the use of multicast Neighbor Solicitation (NS) messages on a network. It does this in two ways:

- **Dropping Multicast Messages:** The feature can block these multicast messages entirely and respond to them on behalf of the intended target.
- **Converting to Unicast:** Alternatively, it can change the multicast traffic into unicast traffic. This means that instead of sending the message to multiple devices at once (multicast), it sends the message directly to a specific device (unicast). To do this, the system replaces the multicast MAC address with a unicast MAC address. This conversion requires the system to know the addresses of devices on

the network and how they are linked to their Layer 2 MAC addresses.

For Wi-Fi scenarios, to increase the efficiency of the NDP process, Neighbor Discovery caching allows the controller to act as a proxy and respond to the NS queries so that it can support address resolution and duplicate address detection. The controller can either respond to an NS on behalf of the wireless client or convert the multicast NS into a unicast one for the target client. Both solutions save wireless resources because the unnecessary delivery to other clients is eliminated.

Summary

In this chapter, you learned the concepts of IP addressing for both IPv4 and IPv6 technologies. You learned about the common DHCP attacks and ways to mitigate them. You also examined the importance of DHCP options like Option 82 and first hop security features of IPv6. Zero trust architecture mandates that every network request is authenticated and authorized, with no implicit trust given to devices or users. DHCP plays a crucial role by assigning and managing IP addresses dynamically; therefore, it is important to safeguard the DHCP process from any kind of attacks to have a solid foundation of zero trust in your environment.

Chapter 6. Automating the Campus

In this chapter, you will learn about the following:

- Campus network automation and its desired outcomes
- Planning as an important part of the campus network automation strategy
- The execution strategy for campus network automation

Overview

Campus networks come in various sizes. Depending on the size of an organization, networks may range from tens of switches to a few hundred. They could serve a few hundred users to as many as a few thousand. When the term *campus network* is mentioned, it is implied that these networks are large. Most, if not all, times the campus is one of the biggest sites an organization has. These sites are typically their corporate headquarters or one of the largest revenue-generating facilities. In some cases, they are a group of buildings that form a large campus space, like universities or corporate parks.

Due to the sheer number of devices that are being deployed to provide the required connectivity, these campuses, in most cases, also house the data centers for the organization. These require connectivity to a local area network (LAN), wide area network (WAN), and data center. These sites are the most critical in the network because of the number of users and endpoints that are connected to the network. Due to the high productivity and the critical nature of these types of sites, they are the crown jewel of an organization.

The biggest challenge on large campus networks is the day-to-day operations and maintenance. If there are set business hours for these sites, most of the maintenance happens after hours or over the weekend. Because the availability of the change windows for these networks is scarce, organizations try to leverage ways that they can be efficient with their regular maintenance and also ensure their hardware is current and performing to the latest industry standards.

Today's modern networks are very flexible. With the introduction of fabric-based architecture in the early 2010s, networks are becoming more agile in terms of serviceability and operations. With newer high-performance devices, organizations are able to leverage virtualization to scale and provide on-demand services without impacting the underlying network. Protocols such as Spanning Tree Protocol (STP) are being removed from the network to provide faster convergence, sometimes a sub-second failover.

When these complex technologies are deployed, there is always a trade-off and a learning curve. In today's modern fabric-based networks, complexity comes in when the network is divided into two layers: underlay and overlay. The underlay provides a highly redundant and resilient routed infrastructure that lays a solid foundation for the overlay networks, which can stretch to multiple edge switches and provide user and endpoint connectivity. While ensuring all the components of the design are configured properly and without any errors, organizations place a high reliance on automation. Also, when automation is leveraged, consistency is guaranteed across the board because the "human error" element is removed when configuring devices manually. To get a campus network up and operational in today's modern network, it is imperative to understand all the aspects of the effort. Just swapping hardware and configurations is not the way to operate anymore. To ensure high productivity and excellent user experience is maintained, a lot of planning takes place. Most importantly, we cannot forget security. This chapter will discuss how to build fully secure, highly scalable, fully automated campus networks.

Planning

Planning is essential for any task. Understanding what is required and what needs to be achieved is crucial to achieving success in small or big projects. Historically, traditional monolithic network refreshes and upgrades involved swapping one type of switch with a newer model with the same configurations. It was not common to clean up residual unused configurations in old devices because operations and network engineers maintaining them needed clear documentation on why or for what purpose that configuration was added. Due to insufficient documentation and a clear design intent for the network, in the past upgrades were often implemented with minimal strategic planning.

Today, many tools provide configuration backups, version control, and other features to ensure the original intent and golden configuration is maintained. However, not all of them help in ensuring regular cleanups are performed on the deployed configuration once they are no longer in use. Modern network orchestrators like Cisco's Catalyst Center have built-in capabilities that perform the task of network configuration deployment as well as configuration cleanup, depending on the intent that has been pushed. This is a great way to ensure network configurations are always current and have the least amount of stale information to maintain performance and serviceability.

The organizations that are embarking on network transformations either in campus or small remote sites are looking into adopting a clean-slate approach. For decades, their networks have grown organically and very rarely in a structured fashion. This unplanned, nonstructured growth has hampered the adoption of new services and flexibility because they have to firefight many issues before the new services are deployed. This also sets them back with time to market because they have to clear any dependencies before actual deployment. In some cases, fast organic growth without proper planning has led the networks to become like a landmine where a small change can have unexpected network convergence or outages, causing even more downtime of the network.

With the clean-slate approach and move toward controller-driven software-defined networks, organizations are taking this one chance to design the ideal network that can not only scale but also be flexible enough to deploy any services they like and also perform optimally. Legacy configurations are no longer relevant because with fabric-based architecture, they need to be removed and rebuilt anyway. With a clean-slate approach, organizations can deploy networks that can run minimalistic configurations. With network controllers managing them, once the intent is pushed, they can not only provision the configurations but also deprovision them when they are no longer in use.

The planning of modern networks has changed dramatically. Figure 6-1 shows the scale of network deployment in time versus the amount of time required to complete major activity milestones. If time for the entire project is considered as 100 percent, planning and design take about 20 percent, implementation about 50 percent, and migration and testing another 30 percent. With campus network automation and controller-driven architectures today, this scale changes. Overall time to deployment and implementation is greatly reduced; however, planning and design now take close to 60 percent of the relative time needed to deploy a network. This figure demonstrates the importance of planning. If done right, implementation and migration will be a breeze. In different terms, this is a "measure twice, cut once" approach.

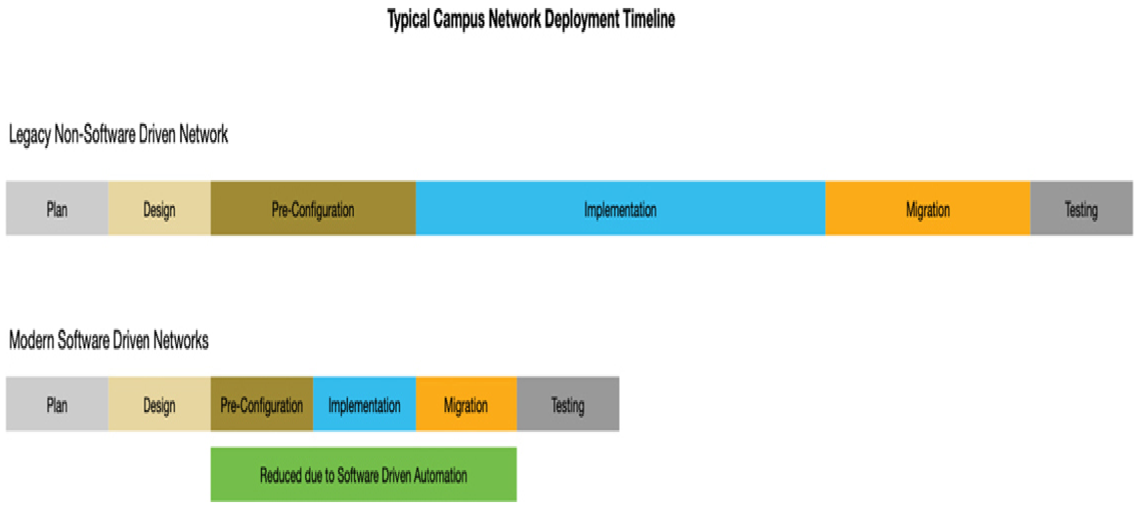


Figure 6-1 *Timeline Percentage of a Typical Network Deployment*

IP Addressing

There can be a whole book written on how to best design an IP addressing schema for an organization. Every device on the network—physical or virtual—needs an IP address. IPv4 addresses are finite and can easily reach their capacity in today's networks. IPv6, on the other hand, is also finite, but due to each address being 128 bits as compared to 32 bits in IPv4, it will be a few decades until we reach an exhaustion phase. Designing IP addresses for a site or an organization requires a holistic view of the network. This is crucial, because if not done correctly, the organization can end up with an inefficient address space with plenty of networks in the global routing table.

In the monolithic network design where all endpoints are in a single flat global routing plane, there are very few options on how to make the address space more efficient. Because everything is in one layer, in most cases, the only option you have is to summarize them. In today's fabric-based access networks, most of the networks are layered with an underlay and overlay architecture. This provides a great advantage in designing an IP addressing schema where the infrastructure IP addresses (underlay of the network) could be summarized or hidden from user IP addresses (overlay of the network). Having a clean routing table will increase efficiency in the network devices as well as operationally. Troubleshooting becomes easier with structured IP addressing; hence, the time to issue resolution and service-level agreements (SLAs) can be maintained. Internet service providers (ISPs) are among the oldest users of fabric-enabled networks such as Multiprotocol Label Switching (MPLS). A lot can be learned from them when it comes to designing an effective IP addressing schema and scaling them. That stands true for both - in the IPv4 and IPv6 world.

For the campus fabric architectures, IP addresses can be divided into three main groups:

- Underlay infrastructure IP addresses
- Management IP addresses
- Overlay user IP addresses

Our sample ISPs have similar groups—infrastructure IP addresses that connect all the SP access, core and backbone routers, management IP addresses to manage all devices—and all of their customer networks are located in the overlay user IP address pools.

Underlay Infrastructure IP Addresses

Underlay infrastructure IP addresses are essentially point-to-point /30 or /31 IP addresses that are being used on Layer 3 links of the infrastructure. These point-to-point IP addresses can either be configured manually, or the entire underlay infrastructure can be deployed using Catalyst Center’s LAN automation feature. These IP addresses are locally contained within the fabric and do not need to be advertised outside of the fabric infrastructure. However, due to the sheer number of point-to-point links, many networks are part of the routing table. Advertising them out in the corporate network is not advisable; hence, a summary of that network supernet can be advertised outside of the underlay network.

At the time of writing, Catalyst Center’s LAN automation feature only supports /30 IP addressing. If /31 point-to-point IP addresses are desired to conserve IPs, building a manual underlay is recommended.

Depending on the business requirements, carving out dedicated supernets for the underlay network is recommended to preserve addressing continuity. Currently, IPv6 in the underlay is not supported, so it would be ideal to utilize RFC 1918 addresses for the underlay. If the requirement is not to advertise the underlay IP addressing to the outside world, the reserved range of 100.64.0.0/10 or 169.254.0.0/16 can be utilized strictly for the infrastructure IP addresses.

Management IP Addresses

Management IP addresses are used for management of the network devices. These IP addresses are /32 and are given to the loopback interface of the router or a switch that is part of the campus fabric. These IP addresses need to be routable and reachable via all infrastructure services. These loopback IP addresses are used by

- Cisco Catalyst Center to manage and configure the devices by enabling SSH connection.

- Control plane nodes to cache information in its mapping database for all the endpoints to the connected device.
- Each node to build Virtual Extensible LAN (VXLAN) encapsulated tunnels between them using these IP addresses as source and destination. The destination IP addresses are provided by the control plane node to the edge and border nodes based on the mapping database entry.
- The Identity Services Engine (ISE) to add this device in its network access device (NAD) database and authorize all AAA authentication and authorization requests.

Since these loopback IP addresses are all /32 IPs, advertising the entire summarized subnet for the site in the global routing infrastructure is highly recommended. This is crucial because any VXLAN tunnels forming from outside of the local SD-Access fabric need to have a nondefault route in the local forwarding information base (FIB) for it to form the VXLAN tunnel. A use case would be by utilizing multi-site remote border (MSRB) or SDA-Transit.

Overlay User IP Addresses

Overlay user IP addresses are the actual endpoint IP addresses that are most commonly used. They are the IP pools that are used for actual production data traffic. These subnets include major VLANs such as data, voice, IoT, guest, and more. It is implied that a user or endpoint subnet needs to be routable across the corporate infrastructure. These subnets are advertised as is based on the size of the network in the respective VRF-aware routing table. Depending on the nature or design of the site, if the site is configured as a stub fabric site (i.e., SD-Access fabric is not a transit to any other type of traffic), in that case, all the individual user IP pools can be summarized into a larger subnet to reduce the prefix count in the corporate or that respective VRF's routing table.

Historically, nonfabric networks relied heavily on Spanning Tree Protocol (STP). STP is great in preventing Layer 2 loops. The downside of this protocol is that it could create a massive Layer 2 broadcast domain. To prevent those broadcasts, as a best practice, most networks were only kept as /24. With monolithic networks, this was not an issue because everything was on a single routing plane. However, this does not scale very well in the fabric-based networks; the reason is that on top of the user subnets, networks are also dealing with underlying infrastructure and management IP addresses. Two of the major advantages of the fabric-enabled networks are the elimination of Layer 2 broadcast and large Layer 2 stretch fabrics. With the use of Layer 3 routed underlay and VXLAN, the need to rely on STP to prevent Layer 2 loops is removed from the equation. There is no way to turn off STP; however, when the routed underlay is used, the STP domain is now restricted to the local switch. This implies, with the elimination of the Layer 2 broadcast, that the extension of Layer 2 can be stretched across the entire campus and user network subnets can be much larger than /24. This, in turn, reduces the number of duplicate networks required in large campuses and makes the routing table more manageable.

This shift in technology has enabled enterprise networks to be designed with more intent and purpose in mind. For example, instead of having multiple VLANs such as data, printer, voice, telepresence, badge readers, and cameras, they can now be combined with similar traffic types like data, voice, and IoT. The user of VXLAN overlay networking technology that encapsulates Layer 2 frames in Layer 3 packets, rather than legacy STP, allows for the creation of large networks. If there is a question about security aspects, that can be addressed via network access control (NAC), security group tags (SGTs), and secure group access control lists (SGACLs). By ensuring right endpoints are tagged with right SGT, SGACLs can be created to prevent communication between devices even in the same VLAN. This paradigm shift can be hard to digest for some organizations that have been building networks in legacy ways for years, but the advantages they could get with this approach are immense, and this is where zero trust comes into play. If every endpoint coming into the network is postured, profiled, and tagged dynamically, creating on-demand policies to restrict traffic becomes a breeze. Most of the high-tech devices being manufactured today come with some form of network connectivity—like coffeemakers and vending machines in cafeterias, or building management systems (BMS) and sensors. Creating different VLANs and scaling them as they grow is not realistic anymore. Therefore, identity-based management is crucial and ever more important. And with IPv6, there will be almost no limit to how big the networks could get.

All the IP pools that are configured in the fabric are dual-stack compatible. They support both IPv4 and IPv6 networks.

Maintaining IP Addressing Continuity

Another big design principle of IP addressing is maintaining IP addressing continuity. Having IP address subnets across the network that cannot be summarized or have no logic in hierarchy is very inefficient. Today's modern network devices have high performance. Some core routers can scale up to a million IP address prefixes in the routing table. However, the routing table is one of the pieces where these broken subnets create problems. These discontiguous IP addresses can cause major complexities in designing effective security policies and traffic engineering. The benefits of maintaining IP addressing continuity are

- IP address summarization
- Supernet-based traffic engineering
- Supernet-based security policies
- Future growth with easy expansion
- Geolocation
- Proper site hierarchy
- Identity and/or user-based network design

Let's look at an example to understand this issue in more detail. Figure 6-2 illustrates a recommended IP addressing structure at a high level and its benefits. At a site level—Site A—there are three major networks; they are the underlay infrastructure, management, and user networks. Taking what we have learned so far, the underlay infrastructure network consists of multiple /30 or /31 prefixes that are local to the site. Management IPs are /32 and are assigned to each of the devices for management and operation of the campus fabric. Finally, the user networks carry all user traffic across the campus infrastructure, but now they are much larger and purpose-driven subnets. Most of the SD-Access sites in deployment today are stub sites. This means these sites do not act as a transit for any through traffic. Any of the backdoor connectivity to other sites or other domains is usually handled via a peer (fusion) router. The various network subnets can be summarized, and essentially, with proper IP address planning, only three subnets need to be advertised out of the fabric to the peer router. This summarization provides a great advantage as compared to a site from legacy network. As more and more sites are brought into the network, the overall routing table becomes more manageable.

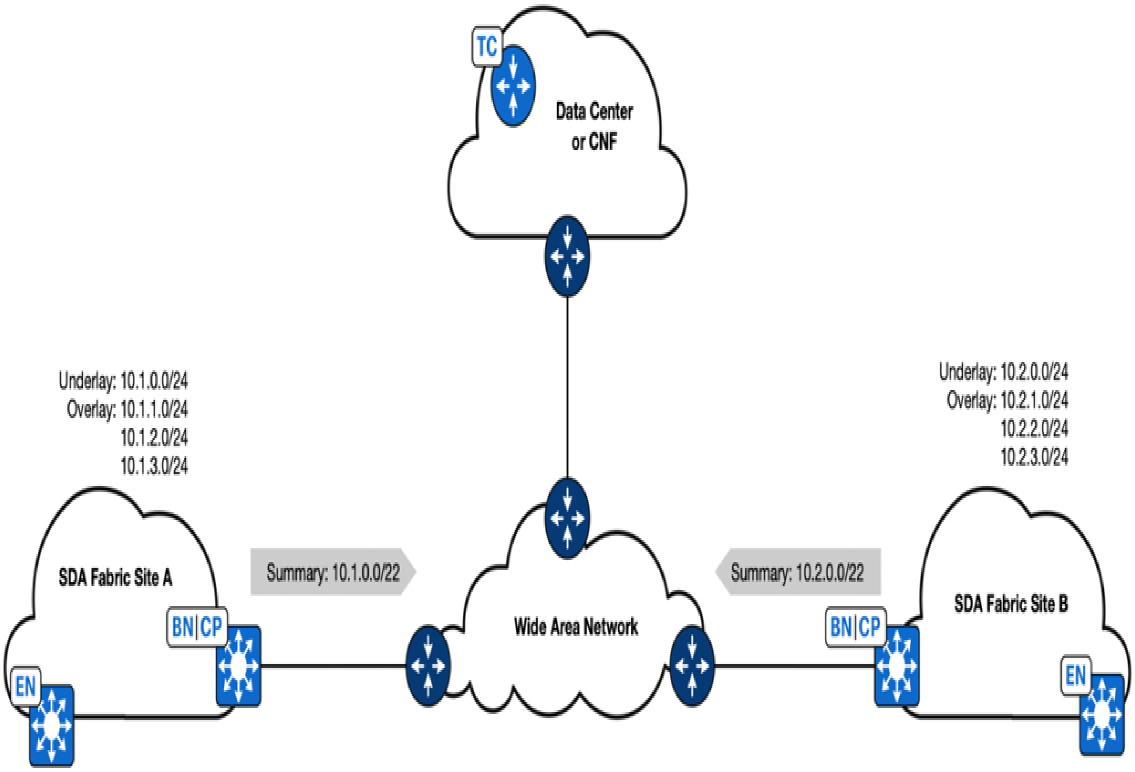


Figure 6-2 *IP Addressing Design*

Another advantage with the separate networks for underlay, management, and overlay is simplified network policies. If there is a need for any access control lists (ACLs) in the network, subnets can be easily identified. This also helps in Day 2 operations: by looking at the issues, support engineers can quickly identify criticality of the subnet and can make decisions regarding any escalation.

Addressing continuity provides an ease in management of IP addresses via Cisco Catalyst Center. As part of the site creation workflow, it becomes much easier to reserve a block of IP addresses at a site level that can be used for various purposes, such as user or infrastructure pools. With IP expansion in mind, additional subnets can be reserved for future expansion of existing networks or building any additional networks that solve new business use cases.

Because one site does not fit all, in an organization, not all sites are of the same size. It is highly recommended to derive a site type template where all the sites of the organization can be placed. This structure will assist in creating a catalog of site types, where this catalog comes with a kit for ease of deployment. The kit would include

- **Physical attributes:**
  - A set(s) of router(s) for WAN connectivity
  - Switches needed for SD-Access fabric
  - Model of wireless LAN controllers (WLCs)
  - Model and base quantity of access points (the final quantity would depend on the site survey)
  - Firewalls as applicable
  - On-premises compute as applicable
- **Logical attributes:**
  - Type of WAN connectivity and transport handoff types (for example, MPLS, Internet, or dark fiber)
  - IP address subnet size
  - Wireless AP group templates
  - QoS and application policies

Having this kit-based architecture simplifies the deployment and also enables certification of each site-type architecture with standardized models and software code versions that are scrubbed and certified by the organization. Any time a site needs to be transformed from a legacy network or needs to be deployed as a net-new site, all that is required is to select the type from the catalog, order the kit hardware, and deploy the automation using kit-specific templates.

Site Hierarchy

Site hierarchy and IP address planning go hand in hand in the design of the corporate network. A structured site hierarchy in Cisco Catalyst Center will assist with accurate reservation of IP pools and avoid any overlaps or gaps in the IP addressing schema. The goal is to have a proper structure that can be utilized for device placement as well as consistent subnet sizes per site types. To illustrate this concept clearly, let's look at Figure 6-3. This figure shows the correlation between a site hierarchy and IP address subnet size and allocation. As we go deeper into the site hierarchy, subnets become smaller but are part of the same global-allocated ranges. This method also provides a clear summarization route from one part of the network to another.

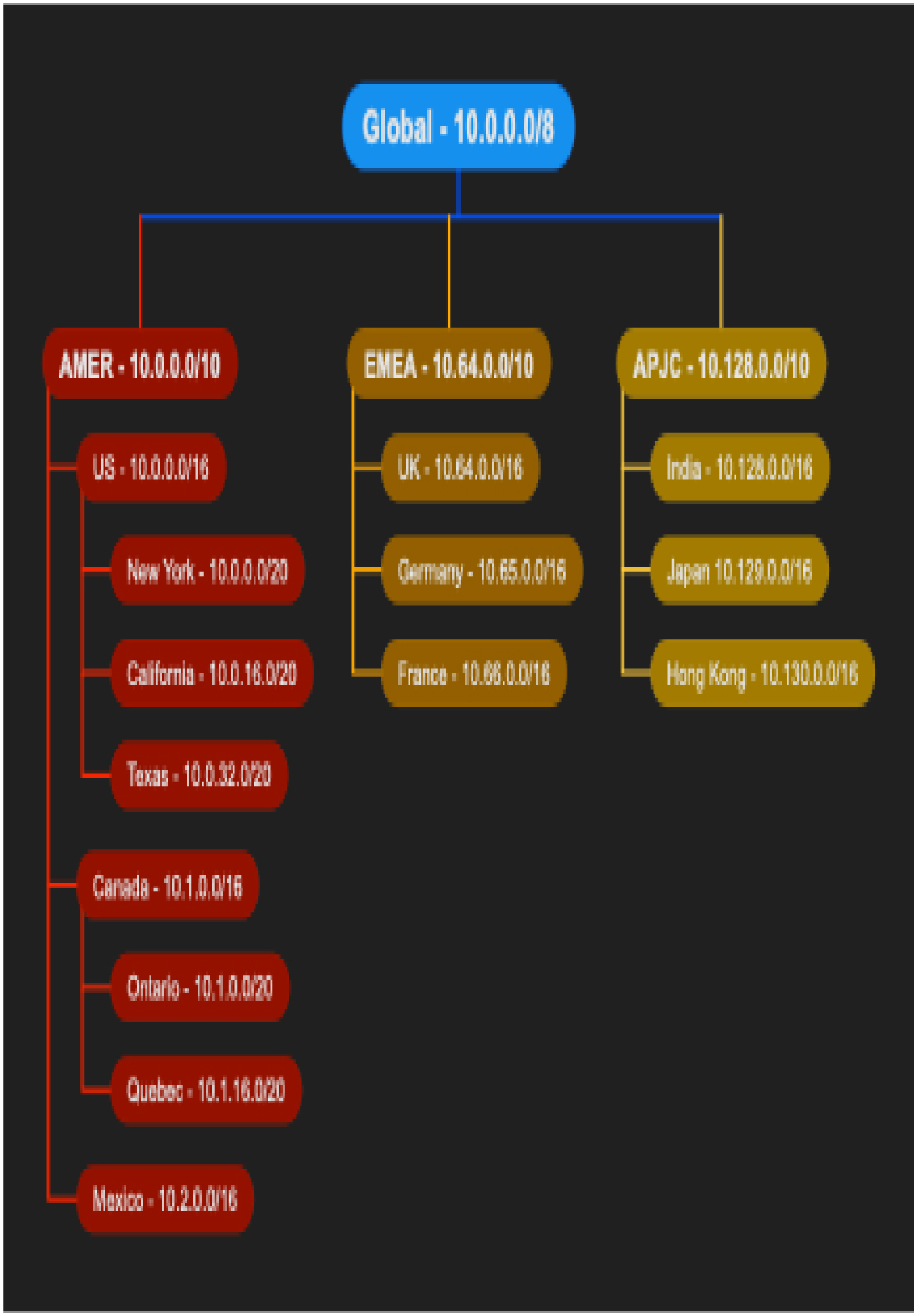


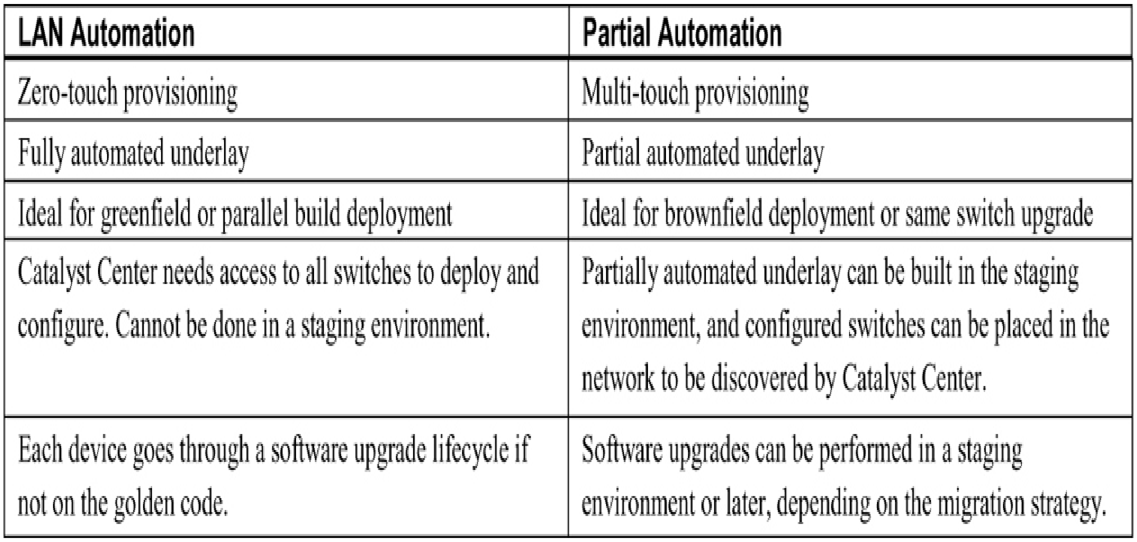
Figure 6-3 *Site Hierarchy and IP Subnet Correlation*

Execution

The site hierarchy is organized and IP addressing has been planned. Now is the time for execution. Network migration execution is a major component of a transformation journey. There are many steps involved in the migration of a site, but for the scope of this chapter and book, we will focus on building the core infrastructure pieces. In this section, we will discuss how to build the underlying infrastructure at scale.

There are many ways to conduct the migration of a site. However, based on our experience, there are two main and most efficient ways to deploy the underlay infrastructure for SD-Access fabric. First is *LAN automation*, and the other is *partial automated deployment*. Both of these methods have their advantages and disadvantages, but both methods have been tried and tested on multiple deployments and have been proven successful and achieve different scale results. Some of the high-level differences between these methods are described in Table 6-1. The details of LAN automation and partial automation are described in subsequent sections.

Table 6.1 *High-Level Differences Between LAN Automation and Partial Automated Deployment*



LAN Automation	Partial Automation
Zero-touch provisioning	Multi-touch provisioning
Fully automated underlay	Partial automated underlay
Ideal for greenfield or parallel build deployment	Ideal for brownfield deployment or same switch upgrade
Catalyst Center needs access to all switches to deploy and configure. Cannot be done in a staging environment.	Partially automated underlay can be built in the staging environment, and configured switches can be placed in the network to be discovered by Catalyst Center.
Each device goes through a software upgrade lifecycle if not on the golden code.	Software upgrades can be performed in a staging environment or later, depending on the migration strategy.

LAN Automation

The LAN automation feature of Cisco’s Catalyst Center revolutionized the way we build massive networks at scale. The idea of zero-touch provisioning (ZTP) of new network devices at scale changes the game on how networks are being deployed today. The term *zero-touch provisioning* means that when a device comes online during its first boot, assuming the network connectivity is provided, it should automatically register to its controller without any key being pressed or command being entered on the device. ZTP of network devices is not new. For years, this technology has been deployed at different parts of the network. Around 2012, we saw ZTP come into small business markets with solutions provided by vendors such as Meraki. The logic was simple: You define the intent on a centralized controller, ship the switches to remote sites, and just plug them in to the network. These switches are factory-shipped with default configurations, which enables them to communicate with the ZTP server. Once they have established the communication, these switches will then get the desired configuration from the network controller and then be fully operational. This process revolutionized deployment for small-to-medium businesses (SMBs) because they did not have to send an IT technician out to all of their remote locations, which was very expensive. They just needed simple instructions and smart hands to help them install the switch and connect the required cables. The rest of the process is done remotely.

This ZTP later evolved in data center and WAN technologies, with ACI and SD-WAN being the biggest users of the process. Again, with the use case, ACI was simple because all the devices were in a single rack-and-row layout, and the APIC controller was usually connected to the fabric itself. Whereas with SD-WAN, all WAN routers would, by default, have WAN connections. That implies they can easily reach the WAN controllers that are located in the cloud via Internet to get their ZTP and configurations automatically.

Coming to campus networks, this process was a little difficult. If we look at the scale of deployment, we are looking at the potential deployment of hundreds of switches, and also various models and port capacities. Getting the individual configuration of those switches ready ahead of time was almost as the same as manually creating them and using a simple copy/paste procedure. Historically, in the traditional Layer 2 networks, with the use of VLAN Trunking Protocol (VTP), the client and server model usually automatically configured VLANs on all LAN switches as soon as they were connected. However, how these VLANs were assigned to the ports and had the right devices in the right VLANs was still a lot of manual work. With fabric-based technologies, the whole process had to be changed. Since there were no Layer 2 VLANs or VTP to be dealt with, a new overhauled process of network automation needed to be designed from the ground up. With the use of Layer 3 routed access layer, and also with multiple campus switch combinations in play (such as high-performance core switches to high-density distribution switches to stackable access layer switches), all combinations of getting the switch ready for the fabric-enabled configuration needed to be taken into account. One of the most critical parts of this architecture is to ensure the workflow is maintained and all components and devices are onboarded into the network.

It is critical to understand what LAN automation accomplishes during the process. The goal of LAN automation is not to build the fabric, but to build the underlay and get it ready so that the SD-Access fabric can be built on top of that. Essentially, LAN automation will ensure a routed access layer is built with the switches in the network with no set device roles. After the LAN automation process is completed, Catalyst Center is not aware of the roles of the devices. They are just in a “managed” state in Catalyst Center without any specific role assigned. Once the LAN automation process is completed, roles such as border, control plane, and edge can be configured as a next step. At a very high level, LAN automation is considered to eliminate the effort required for hardening the basic device, using the underlay IP addressing schema, configuring the routing protocol, and lastly, employing the underlay multicast configuration. LAN automation provides zero-touch provisioning, end-to-end topology, resilience, security, and compliance.

LAN Automation Process and Workflow

LAN automation involves a series of tasks that are completed in a specific order to achieve the desired outcome. The overall goal is to build a fully compliant routed access network. *Fully compliant* in this instance means that the devices are all upgraded with golden code, all links are configured as point-to-point Layer 3 links, and all the devices that are part of the LAN automation process are ready and in a “managed” state within Catalyst Center. Once the underlying infrastructure is built, devices with any role such as edge or border/control plane node can be configured from Catalyst Center.

At the time of writing this chapter, LAN automation accomplishes the many tasks. These tasks may change in the future as the feature evolves; therefore, it is highly recommended that you refer to the latest Cisco documentation for a step-by-step guide.

> **Note**
>
> LAN automation uses zero-touch provisioning. That implies any device that is part of the process needs to run the Plug and Play (PnP) process. If, during the bootup process, any of the devices are interrupted via the CLI, the PnP process on that device will fail. To reset the configuration back to the factory settings, you should add the commands shown in Examples 6-1 through 6-3 to the affected switches.

Example 6-1 *CLI Config Mode*

```
no pnp profile pnp-zero-touch
no crypto pki certificate pool
Also remove any other crypto certs shown by "show run | inc c
crypto key zeroize
config-register 0x2102 or 0x0102 (if not already)
do write
end
```

Example 6-2 *CLI Exec Mode*

```
delete /force nvram:*.cer
delete /force stby-nvram:*.cer (if a stack)
delete /force flash:pnp-reset-config.cfg
```

```
    write erase
    reload (enter no if asked to save)
```

For Cisco IOS XE 16.12.x or later, use the code in Example 6-3.

Example 6-3 *Cisco IOS XE 16.12x or later CLI Excec Mode*

```
pnp service reset no-prompt
```

LAN automation has four main stages: plan, design, discover, and provision.

Planning Stage

In the planning stage, a seed device is selected for the network. As the name suggests, the seed device is used as a root to discover the topology and configure its connected devices and get them ready for the fabric underlay. The seed device can be either onboarded into the network via Plug and Play or be configured manually; its main goal is to provide full connectivity and manageability to Cisco Catalyst Center so that other devices downstream can be configured.

There are limitations on how many levels can be connected for LAN automation. At the time of writing, nodes more than two levels are not supported as part of the LAN automation process. That is, if the seed device is a core device, connected distribution and access layer switches are supported; however, any extension of access layer switches, such as extended nodes or other daisy-chained switches, are not supported as part of the LAN automation process. For such layers, a multistep LAN automation process can be executed. In this process, once the first pass of LAN automation is configured, another device such as a distribution or an access layer switch can be selected as a seed device, and more switches downstream can be configured via the LAN automation process.

LAN automation assigns all devices a loopback IP address and also configures all interconnected links as Layer 3 point-to-point. During the LAN automation process, a dedicated IP address subnet needs to be reserved as a LAN automation pool that will be used to carve out loopback and point-to-point IP addresses. As discussed earlier in this chapter, it is highly critical to plan out growth and the number of IP addresses that are required for this process. During the LAN automation process, one part of that IP address is used as a temporary DHCP pool. This is for any new devices coming onboard via PnP and will get a temporary IP address from this DHCP pool. The second part of that subnet is carved out for loopback IP addresses—specifically, loopback 0 and loopback 6000. These IP addresses are assigned to the devices in sequence as they are onboarded in order via the PnP process. The third part of the IP subnet is reserved for point-to-point links between the devices. These IP addresses are /31 for point-to-point and /32 for loopbacks. Reserving a minimum subnet size of /24 for sites with LAN automation is highly recommended.

Design Stage

The design stage of the LAN automation process is focused on setting up the correct site hierarchy for the devices that are being LAN automated. This is essential because all the settings that are required for that site in terms of device hardening are pushed and set up from the beginning. This stage involves the following:

1. Creating or building a global site structure
2. Configuring global and local network services
3. Configuring global device credentials (for first site LAN automation process)
4. Designing a global IP address pool and assigning the LAN automation pools from the global IP address pools

Discovery Stage

The discovery stage allows you to discover the seed device and get it ready for the LAN automation stage. In this stage, a new discovery is created for that site, and the seed device is being discovered. Once the devices are discovered, they can be assigned to the site and provisioned via Catalyst Center. Once they are provisioned, they are in a “managed” state, which allows Catalyst Center to push configuration to the devices. These configurations could be related either to the LAN automation process or any future fabric-related configuration.

Provisioning Stage

The provisioning stage is where the biggest change in the network happens. The LAN automation process is initiated in this stage, and the switches are built and prepped for the fabric deployment. Some prerequisites need to be accounted for. Before the LAN automation process is started, a validation needs to be carried out to check whether the underlay subnet of the LAN automation (which has not been deployed in the network yet) has a route back to the Catalyst Center as well as the rest of the internal network. This /24 LAN automation subnet needs to be routed via an IGP, EGP, or redistributed via a static route to so that once Catalyst Center runs LAN automation, it will maintain the connectivity. Besides the route reachability, the following points need to be taken into consideration:

- Ensure that the management port is unplugged for the devices that are part of the LAN automation process.
- Ensure that all the seed device ports that are connected to the downstream devices are in Layer 2 mode.
- Ensure that a primary seed port does not block Spanning Tree Protocol (STP).
- Ensure that the devices that are part of the LAN automation are not present in the Catalyst Center inventory. If they are from a stale configuration, remove them.
- Ensure that the devices are not present in the current PnP process of the Catalyst Center.
- Ensure that a DNA-Advantage license is used for the devices that are part of LAN automation.
- Ensure that the PnP agents on the devices that are part of the LAN automation are in INSTALL mode.

Once the prerequisites steps are verified, LAN automation workflow can be started from Catalyst Center from the device. The LAN automation workflow provides step-by-step information on features that need to be enabled and interfaces that need to be selected for the automated provisioning. Once the values are entered and the LAN automation process is started, the configuration shown in Example 6-4 is pushed down to the seed device by Catalyst Center.

Example 6-4 *LAN Automation Seed Device Configuration*

```
!exec: enable
!
system mtu 9100
!
ip multicast-routing
ip pim ssm default
!
Loopback IP and IS-IS configuration. (If the secondary seed is co
loopback IP and IS-IS configuration.)
interface Loopback0
   ip address 10.4.210.123 255.255.255.255
   description Fabric Node Router ID
!
router isis
   net 49.0000.0100.0421.0123.00
   domain-password *
   ispf level-1-2
  metric-style wide
  nsf ietf
   log-adjacency-changes
   bfd all-interfaces
   passive-interface Loopback0
   default-information originate
!
interface Loopback0
ip router isis
```

```
clns mtu 1400

ip pim sparse-mode
exit
!
DHCP pool information:
ip dhcp pool nw_orchestration_pool
  network 10.4.218.0 255.255.255.192
  option 43 ascii 5A1D;B2;K4;I10.4.249.241;J80;
  default-router 10.4.218.1
  class ciscopnp
    address range 10.4.218.2 10.4.218.62
!
ip dhcp class ciscopnp
  option 60 hex 636973636f706e70
!
ip dhcp excluded-address 10.4.218.1
!
VLAN 1 configuration:
vlan 1
!
interface Vlan1
  ip address 10.4.218.1 255.255.255.192
  no shutdown
  ip router isis
  clns mtu 4100
   bfd interval 500 min_rx 500 multiplier 3
   no bfd echo
exit
!
Switch port configuration on interfaces used for discovery. (Each
device gets this configuration.)
```

```
interface TenGigabitEthernet1/1/8
  switchport
  switchport mode access
  switchport access vlan 1
!
interface TenGigabitEthernet1/1/7
   switchport
   switchport mode access
  switchport access vlan 1
exit
Multicast configuration (optional; only configured if the multica
If the Rendezvous Point (RP) for the underlay multicast needs to
automation with multicast enabled using border switch as the seed
If the peer seed is configured, these multicast CLIs are pushed o
is used to configure Loopback0 on both the primary and peer seeds
interface Loopback0
  ip address 10.4.218.67 255.255.255.255
  ip pim sparse-mode
  ip router isis

ip pim register-source Loopback0
ip pim rp-address 10.4.218.67
```

Note

The IP addresses in the configuration shown in Example 6-4 are examples and would vary depending on the network where the LAN automation process is run.

Once this LAN automation process has started, all the remote devices that are part of the LAN automation process will kick off their PnP agent and will get an IP address for the temporary DHCP server from the seed device, and also with option 43 and 60 of DHCP, they will be able to reach the Catalyst Center to get the rest of the configurations. Once all the devices have been discovered in the PnP process, the LAN automation process can be stopped. During that stage, Catalyst Center will push the final configuration with Layer 3 interfaces and loopbacks to all the switches and reload them. For all the devices that were being discovered, LAN automation will show them as “completed,” which means that these devices can now be provisioned and configured for their respective device role.

LAN automation is great for getting the underlay built for large greenfield deployments. If the site that has been LAN automated is in need of an expansion, the process can be run again on the new devices with new seed devices, and those devices can be onboarded as well. Overall, in the journey of campus automation, LAN automation plays an important role in taking a step further in software-driven architecture.

API-Based LAN Automation Provisioning

Taking a step further with LAN automation, the process discussed thus far involves invoking a workflow on the Catalyst Center UI and walking through the process of providing the values and initiating the LAN automation process. Today, with the newer releases of Cisco Catalyst Center, LAN automation APIs are available to consume, and a custom LAN automation tool or a script can be built to take automation even further. API-based Ansible modules are available on Ansible today. Cisco has released Catalyst Center SDKs with these API modules. These APIs are intent driven and can be run based on the requirements. Example 6-5 shows a LAN automation API parameter.

Example 6-5 *LAN Automation API Parameter YAML File*

```
- name: Create
  cisco.dnac.lan_automation_create:
    dnac_host: "{{dnac_host}}"
    dnac_username: "{{dnac_username}}"
    dnac_password: "{{dnac_password}}"
    dnac_verify: "{{dnac_verify}}"
    dnac_port: "{{dnac_port}}"
    dnac_version: "{{dnac_version}}"
    dnac_debug: "{{dnac_debug}}"
    payload:
    - discoveredDeviceSiteNameHierarchy: string
      hostNameFileId: string
      hostNamePrefix: string
      ipPools:
      - ipPoolName: string
        ipPoolRole: string
      isisDomainPwd: string
      mulitcastEnabled: true
      peerDeviceManagmentIPAddress: string
      primaryDeviceInterfaceNames:
      - string
      primaryDeviceManagmentIPAddress: string
      redistributeIsisToBgp: true
```

You can find more Catalyst Center APIs for LAN automation in the latest documentation at https://developer.cisco.com.

Partial Automated Deployment

At this point, you should understand how LAN automation can make a massive impact on bringing up a new deployment. With the ease of zero-trust provisioning and getting devices up to code and basic configuration, you can get a jump-start on the network deployment. This is massive in terms of time to market. However, not all buildings or sites in an organization are *net new*. In a majority of the cases, brownfield migration is required due to the following:

- There is a lack of space for parallel build in the intermediate distribution frame/main distribution frame (IDF/MDF) closets.
- Edge devices are already upgraded to newer models and their role needs to be changed from legacy to fabric enabled.

- An organization is working with a partner to procure and pre-stage the devices at a central warehouse before the devices are shipped to the site. This process does the following:
  - Checks for any potential dead on arrival (DoA)
  - Performs software upgrades
  - Deploys initial configurations
  - Connect scables and optics
  - Prepares the Method of Procedure (MoP)

If LAN automation is to be performed, all of these pieces warrant access to Catalyst Center and also high bandwidth to download golden image files to the switches and then provision them. If the devices are not in the customer's facility, getting full administrator access of the production Catalyst Center to partner is not viable. Consequently, alternate methods are used to achieve similar results but at scale. The following process describes one of the major use cases that we have observed in our deployments where an organization is trying to upgrade its hardware and move to SD-Access at the same time.

The process of partially automated deployment is as follows:

1. Upgrade devices to the golden code offline via USB.
2. Pre-stage minimal required configurations on the devices.
3. Ship the devices to the site.
4. Unbox the devices at the site, rack and stack them, connect the uplink/downlink cables, and power on.
5. Once reachability to Catalyst Center is established, run a discovery, provision, manage, and assign fabric roles to them.

These steps will essentially assist with breaking up the device onboarding process for scale. When the business needs to perform a massive organizationwide upgrade of devices and solutions in a short amount of time, an assembly-line approach might be required. This process can be broken down into multiple teams where one team is responsible for staging the devices and another is responsible for bringing those devices into SD-Access fabric. After step 5, the processes for LAN automation and partial automation deployment are the same.

For step 2, the minimal configuration required to be deployed on the devices is shown in Example 6-6.

Example 6-6 *Minimum Base Underlay Configuration*

```
ip routing
!
hostname <HOSTNAME>
!
no aaa new-model
!
archive
 log config
  logging enable
  logging syslog content plaintext
!
username dnac privilege 15 secret <SECRET>
username sdaadmin privilege 15 secret <SECRET>
!
no ip domain lookup
ip domain name <DOMAIN_NAME>
!
crypto key gen rsa gen mod 2048
!
interface loopback 0
 description DNAC Mgmt Interface
 ip address <IP_ADDRESS>
 ip router isis
!
interface <P2P_LINKS>
 no switchport
 ip address <IP_ADDRESS>
 ip router isis
 bfd interval 300 min_rx 300 multiplier 3
!
router isis
 net <NET_ADDRESS>
 is-type level-2-only
 metric-style wide
 log adj changes
 bfd all-interfaces
!
snmp-server community public RO version 2c
snmp-server community private RW version 2c
!
line vty 0 15
 login local
 transport input ssh
!
```

The benefit of this approach lies with one of the most commonly seen scenarios. In this scenario, when a device needs to be hot-swapped with a new configuration and hardware, it becomes much easier to check the hardware, upgrade the code, and stage it with the preconfigurations. When the time comes to swap the device during the maintenance window, it becomes easier to replace, connect power and uplinks, discover and provision the device via Catalyst Center, and then connect all the endpoint connections. With dot1x enabled, the old devices would onboard the network, and for any special devices that had static IPs or no dot1x capabilities, those ports can be configured via Catalyst Center on that switch. With this migration use case, the network turnaround time is faster because this can be done per IDF basis, and multiple IDFs can be completed at the same time too. And with lack of extra power, rack, or uplink connections, this becomes the best solution to migrate to the new network.

Another major scenario is a two-stage migration. This approach is not so common now, but it was during the earlier stages of SD-Access deployment. We have witnessed organizations taking on two-stage migration where they would upgrade their legacy switching infrastructure with newer SD-Access[nd]capable hardware, and then once the entire site was upgraded, they would convert that to SD-Access. The biggest challenge we observed with this method was taking down the site two times for two different maintenance windows. This was a challenge for multiple business units, especially if a site had a 24x7 operation, such as a manufacturing facility or a hospital. Outages are not taken lightly a time for leisure at these locations. A most common use case for this approach is when the legacy hardware is out of support and an entire new SD-Access design cannot be deployed in an extremely short amount of time. This approach gets an organization to a supported hardware first, which paves the way for a software-driven architecture. If, for any reason, there is a need to have such a migration approach, the following strategy can be employed:

1. Replace the legacy hardware with a like-for-like configuration from the old device (code permitting).
2. On moving to an SD-Access strategy, pre-generate a base underlay configuration based on Example 6-4.
3. Deploy this base underlay configuration to all the respective deployed switches.
4. On the day of the maintenance window to convert them to SD-Access, save the legacy configuration to a switch's hard disk, and reboot the switch using the base underlay configuration as a startup configuration.
5. Once the device reboots using the new base underlay configuration, make appropriate IP address changes on the upstream switch and then discover and provision this switch from Catalyst Center.

The strategies mentioned here can be used based on the needs of the organization. These strategies are not exclusive. Depending on the size of the site or building, one or more strategies can be deployed on the IDF to achieve complete migration to a software-driven architecture.

Summary

In this chapter, you learned what campus network automation means. You learned about the importance of planning the entire architecture, where planning the IP addresses of a site is the most critical. You looked at different strategies for IP address planning, site hierarchy, and execution strategies. In the later sections of the chapter, you examined the LAN automation process, workflow, and its deployment strategies. You also looked into how partially automated networks would help in migration of brownfield networks.

References

1. LAN automation: https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/dna-center/tech_notes/b_dnac_sda_lan_automation_deployment.xhtml
2. API-based Ansible modules available on Ansible: https://docs.ansible.com/ansible/latest/collections/cisco/dnac/lan_automation_create_module.xhtml
3. Catalyst Center APIs for LAN automation: https://developer.cisco.com/docs/dna-center/#!lan-automation-status-by-id

Chapter 7. Plug-and-Play and Zero-Touch Provisioning

In this chapter, you will learn about the following:

- How an operator can onboard new nodes into the network
- Differences between Plug-and-Play and zero touch provisioning
- Meraki onboarding process and provisioning requirements
- How to use APIs within Meraki and Catalyst Center to onboard devices

Overview

Over the years, the need to scale, expand, and deploy IP networks securely in the most remote locations on the planet has changed from a dream to a reality. With the deployment of fiber optic cabling, and thanks to improvements in radio frequency allocation and microwave-based radio backhaul technologies, even the most remote users have been able to embrace access to the Internet, plus corporate and public service networks.

One of the key enablers in this transformation that provided the capability to bring initial devices quickly, securely, and relatively seamlessly into service has been *Plug-and-Play (PnP)*, or as many detractors prefer to call it *Plug-and-Pray*. The Plug-and-Play capabilities in the Cisco product line were first introduced as a progression from Auto-install, which became a default part of the IOS product line. It was initially released for use with legacy routers in 1993 and expanded in 2004 with support for local area network (LAN) interfaces in conjunction with Dynamic Host Configuration

Protocol (DHCP), opening the door to more elaborate use cases through a software technology train IOS version.

Auto-install was an early capability provided in IOS software to allow for Trivial File Transfer Protocol (TFTP) values specifying configuration images to load on routers and switches. Although it was useful at the time, it came with inherent limitations that impeded it from being able to scale appropriately, which largely limited its usage to local network activities or, in many cases, evergreening or rebuilding activities for dynamic lab environments. These limitations were largely attributed to the insecure transfer of configurations via TFTP and inflexibility in terms of device allocation to configuration, which required provisioning.

In 2013, Plug-and-Play was introduced into the product line orchestrated by Cisco Prime Infrastructure, which was one of Cisco's earlier network management platforms that was heavily used in the networking industry prior to the release of newer orchestrators, such as APIC-EM and eventually Cisco Catalyst Center. Cisco Prime Infrastructure provided a new and rich set of functionality and features. For example, it allowed not only the initial deployment of a network node via DHCP and/or DNS-based discovery but also the ability to accomplish more complex workflows, such as achieving rollback scenarios if the Plug-and-Play process stalled or failed during initial installation.

Having these resilience capabilities available opened the door for large-scale organizations and service providers to turn to PnP to lower their deployment costs for a broad range of offerings, providing the value proposition of not requiring skilled network technicians to be present in remote locations to configure routers and switches over the console. This functionality has become a financially lucrative and operationally beneficial consideration. Through the use of this technology, Cisco has worked with many customers to rapidly scale up large deployments. Examples include financial institutions with 5000+ branch locations, service providers sending new customer-premises equipment (CPE) to customer sites, and technology companies deploying home office routers for tens of thousands of employees.

In this chapter, we will explore the technical foundations of the Plug-and-Play solution and some real-world scenarios where the use of Plug-and-Play provided a more simplistic means to deploy and scale large-scale network environments.

Plug-and-Play Provisioning

When you're deciding to deploy Plug-and-Play within your network environment, the first decision that you need to make is how the network will inform the to-be-onboarded devices that they should execute Plug-and-Play onboarding with a particular server. This issue is often the topic of long discussions with customers during the project planning phase, because existing solutions may leverage common mechanisms such as DHCP Option 43 or the A-Record for PnP servers existing in the network already. In the following sections, we will explore the different options available for use and the pros, cons, and nuances associated with the respective option that is selected.

Cisco Catalyst Center Call Flow

To ensure that the right foundation is present when talking about Plug-and-Play, let's begin by looking at how a successful Plug-and-Play communication takes place from beginning to end. The key components to the communication flow are the *PnP agent* and the *PnP server*. The agent represents a piece of software running on the device executing the Plug-and-Play process—for instance, a router or switch.

In enterprise networks, Cisco Catalyst Center provides the means to perform network automation and assurance capabilities. Following the success of Cisco Prime Infrastructure, the platform provides improved capabilities in the domain or structured automation, intent-based APIs, and onboarding capabilities such as Plug-and-Play server functionality.

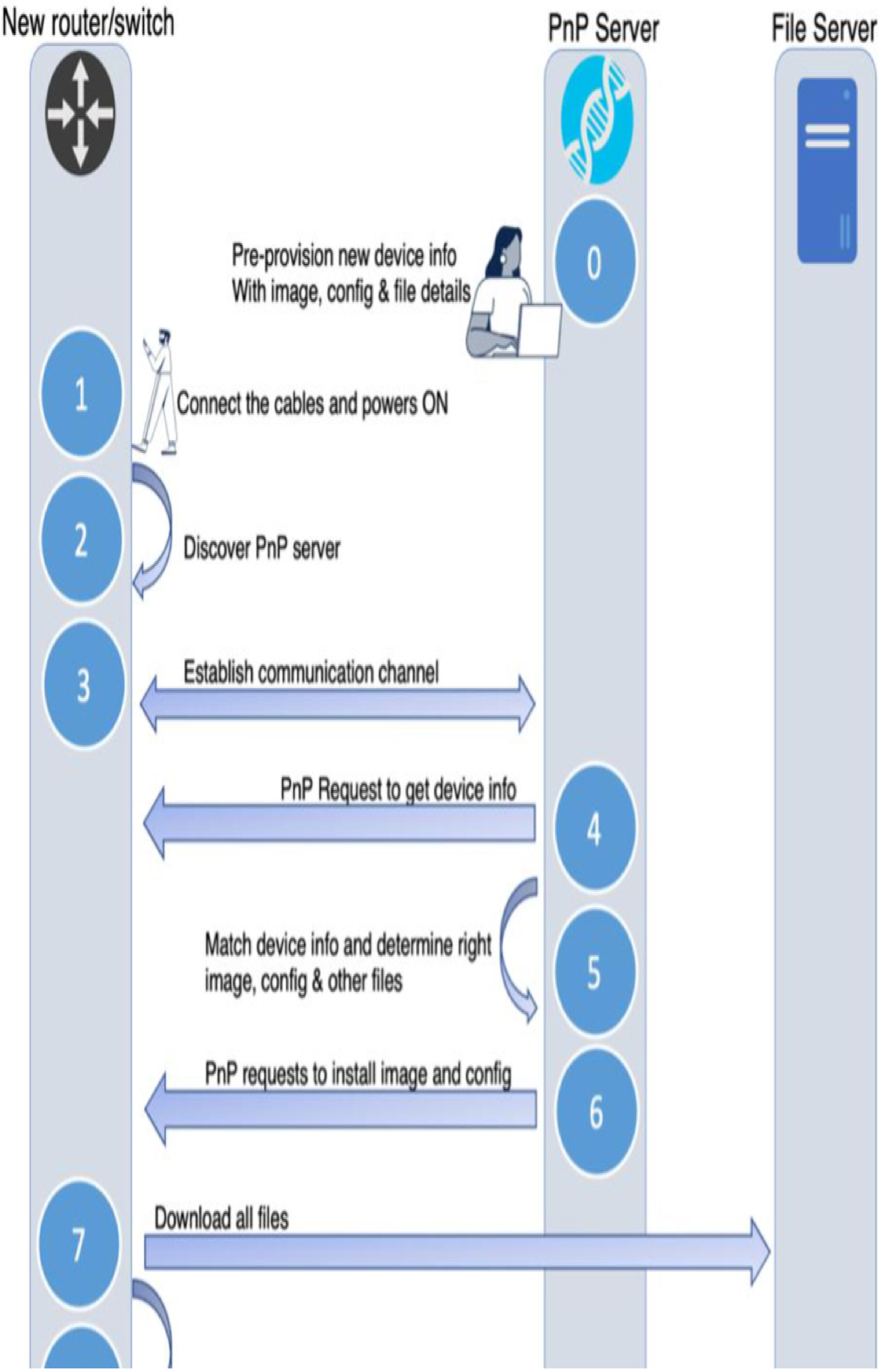
Cisco has both observed and supported many customers using the automation and Plug-and-Play deployment flows to bring up massive campus environments with thousands of switches. Cisco also has supported customers using Catalyst Center to deploy, roll out, and scale critical global sites for both branch office and campus constructs.

During startup, the Cisco network device will transition through a discovery sequence that steps through the following options to identify a viable PnP server:

- DHCP Option 43
- DNS (pnpserver.xxxxx)
- PnP connect (pnpconnect.cisco.com)

The PnP server represents the system that is responsible for the provisioning of configuration, software, and licensing to the agent devices.

A basic example of the communications between agent and server is illustrated in Figure 7-1.


New router/switch
PnP Server
File Server
0
Pre-provision new device info
With image, config & file details
1
Connect the cables and powers ON
2
Discover PnP server
3
Establish communication channel
PnP Request to get device info
4
Match device info and determine right
image, config & other files
5
PnP requests to install image and config
6
7
Download all files

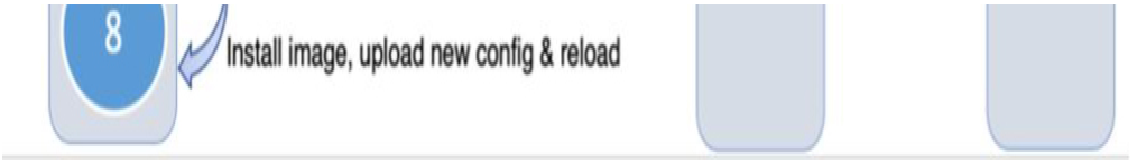


Figure 7-1 *PnP Onboarding Process*

Certificates

The use of certificates in the context of Plug-and-Play ensures a secure exchange of information when communicating between an onboarding device and the PnP server; the certificate itself can exist in multiple forms. On the Plug-and-Play server, server certificates are typically issued by a private certificate authority (CA). To ensure that the certificate validation can properly take place, in most systems participating in Plug-and-Play, it is important that either the fully qualified domain names (FQDNs) associated with the resolved server and/or the IP addresses are included within the Subject Alternative Name fields.

Example 7-1 shows the format that is normally expected for a web server[nd]based certificate utilizing both FQDN and IP addressing.

Example 7-1 *Catalyst Center Web Certificate*

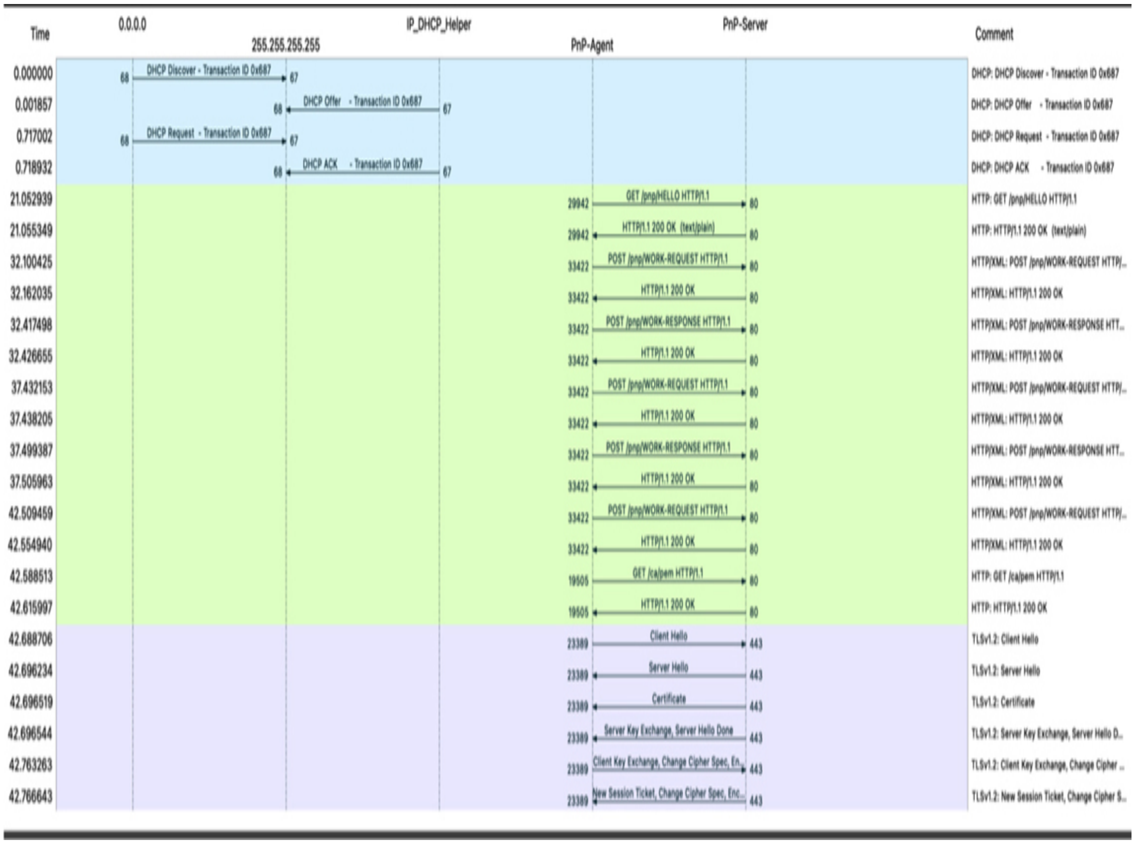
```
unix-workstation$  openssl x509 -inform pem -noout -text -in 'cc-
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            70:00:00:00:49:e8:fe:ab:ca:9e:03:02:18:00:00:00:00:00
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: DC=com, DC=cisco, DC=muc07cxlab, CN=muc07cxlab-PD
        Validity
            Not Before: Feb 25 13:27:39 2022 GMT
            Not After : Feb 25 13:27:39 2024 GMT
        Subject: CN=cc-1.muc07cxlab.cisco.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
```

```
                RSA Public-Key: (2048 bit)
                Modulus:
                    <REMOVED FOR BREVITY>
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, TLS Web Server Aut
            X509v3 Subject Alternative Name:
                DNS:cc-1.muc07cxlab.cisco.com, DNS:pnpserver.garc
ent.muc07cxlab.cisco.com, IP Address:172.16.0.74, IP Address:172.
<REDACTED FOR BREVITY>
```

Within the certificate X509 Subject Alternative Name field, you can see both the FQDN entries associated with the Plug-and-Play server, in addition to the IP addresses associated with this server.

When a Plug-and-Play agent attempts to validate the trustworthiness of a Plug-and-Play server, it verifies that the FQDN or the IP address that it received during the discovery process matches up with the common and/or subject alternate names that are present in the certificate that is in use. It uses the retrieved certificate that is requested in the second REST HTTP GET request, as shown in Figure 7-2.

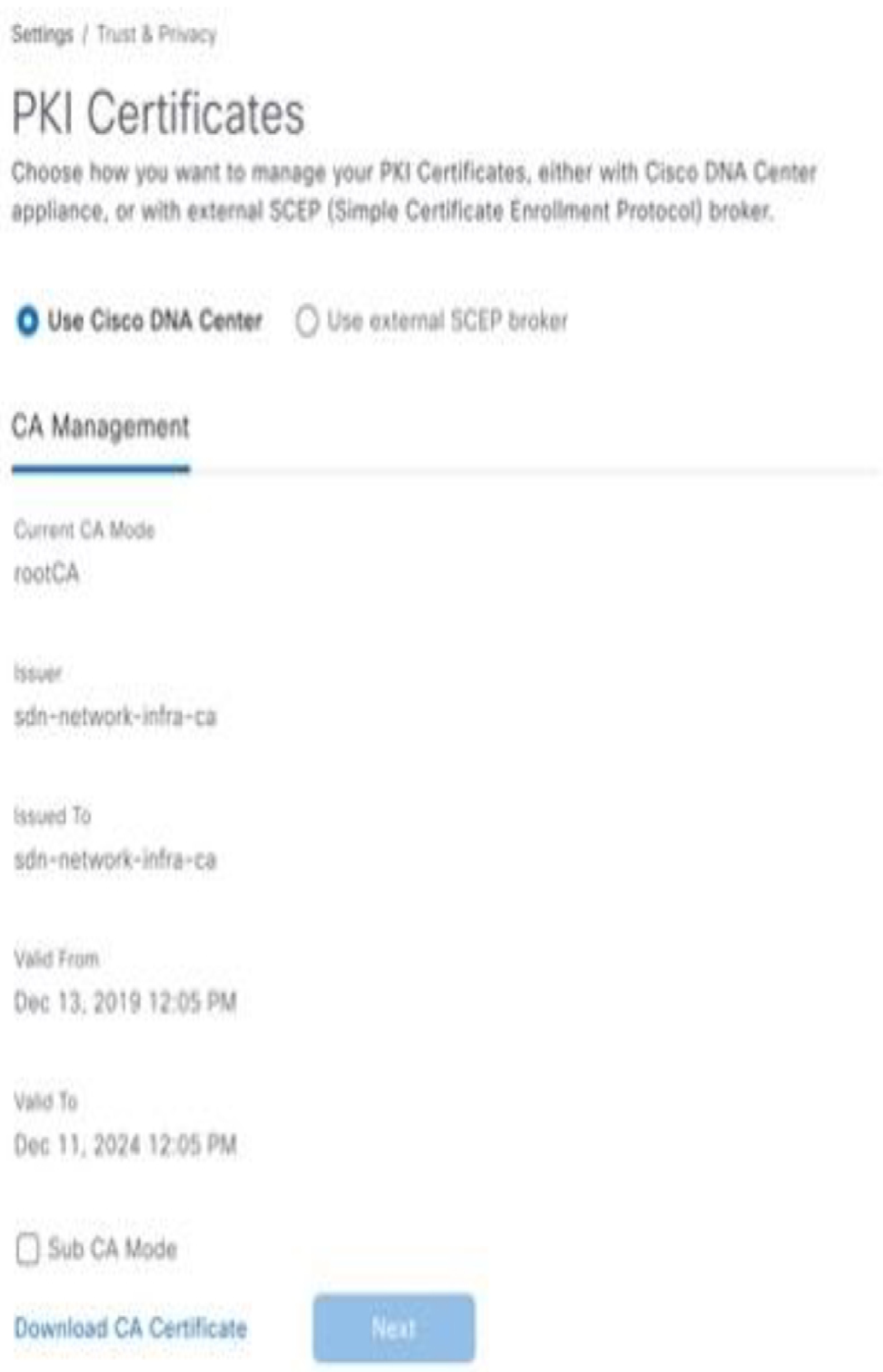
This activity takes place prior to the system transitioning to leverage an HTTPS connection. Once the HTTPS connection is established, a further device-specific certificate is issued to the PnP agent device from the Plug-and-Play PKI infrastructure, which is hosted on the Catalyst Center Server.



Time	Comment
0.000000	DHCP: DHCP Discover - Transaction ID 0x687
0.001857	DHCP: DHCP Offer - Transaction ID 0x687
0.717002	DHCP: DHCP Request - Transaction ID 0x687
0.718932	DHCP: DHCP ACK - Transaction ID 0x687
21.052939	HTTP: GET /pnp/HELLO HTTP/1.1
21.055349	HTTP: HTTP/1.1 200 OK (text/plain)
32.100425	HTTP/XML: POST /pnp/WORK-REQUEST HTTP/...
32.162035	HTTP/XML: HTTP/1.1 200 OK
32.417498	HTTP/XML: POST /pnp/WORK-RESPONSE HTT...
32.426655	HTTP/XML: HTTP/1.1 200 OK
37.432153	HTTP/XML: POST /pnp/WORK-REQUEST HTTP/...
37.438205	HTTP/XML: HTTP/1.1 200 OK
37.499387	HTTP/XML: POST /pnp/WORK-RESPONSE HTT...
37.505963	HTTP/XML: HTTP/1.1 200 OK
42.509459	HTTP/XML: POST /pnp/WORK-REQUEST HTTP/...
42.554940	HTTP/XML: HTTP/1.1 200 OK
42.588513	HTTP: GET /ca/pem HTTP/1.1
42.615997	HTTP: HTTP/1.1 200 OK
42.688706	TLSv1.2: Client Hello
42.696234	TLSv1.2: Server Hello
42.696519	TLSv1.2: Certificate
42.696544	TLSv1.2: Server Key Exchange, Server Hello D...
42.763263	TLSv1.2: Client Key Exchange, Change Cipher ...
42.766643	TLSv1.2: New Session Ticket, Change Cipher S...

Figure 7-2 *Wireshark PnP Flow Graph*

In Figure 7-3, you can see an overview of the Catalyst Center PKI CA, which corresponds to the device certificate allocation. Optionally, the Catalyst Center can be configured as a sub-CA to a third-party certificate authority.



Settings / Trust & Privacy

PKI Certificates

Choose how you want to manage your PKI Certificates, either with Cisco DNA Center appliance, or with external SCEP (Simple Certificate Enrollment Protocol) broker.

● Use Cisco DNA Center ○ Use external SCEP broker

CA Management

Current CA Mode
rootCA

Issuer
sdn-network-infra-ca

Issued To
sdn-network-infra-ca

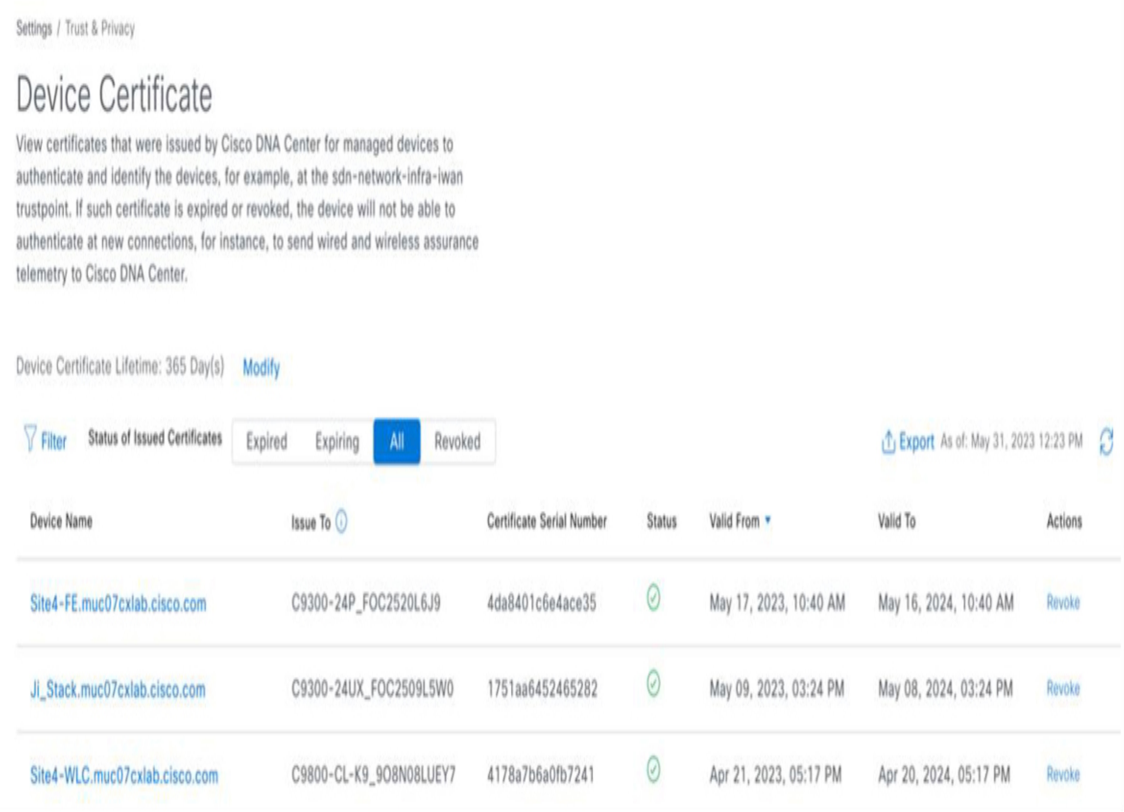
Valid From
Dec 13, 2019 12:05 PM

Valid To
Dec 11, 2024 12:05 PM

□ Sub CA Mode

Figure 7-3 *PnP Device Certificate Authority Settings*

Allocated certificates from the Catalyst Center CA during the Plug-and-Play process are visible in the system settings within the **Trust & Privacy** section, as shown in Figure 7-4.



Settings / Trust & Privacy

Device Certificate

View certificates that were issued by Cisco DNA Center for managed devices to authenticate and identify the devices, for example, at the sdn-network-infra-iwan trustpoint. If such certificate is expired or revoked, the device will not be able to authenticate at new connections, for instance, to send wired and wireless assurance telemetry to Cisco DNA Center.

Device Certificate Lifetime: 365 Day(s) Modify

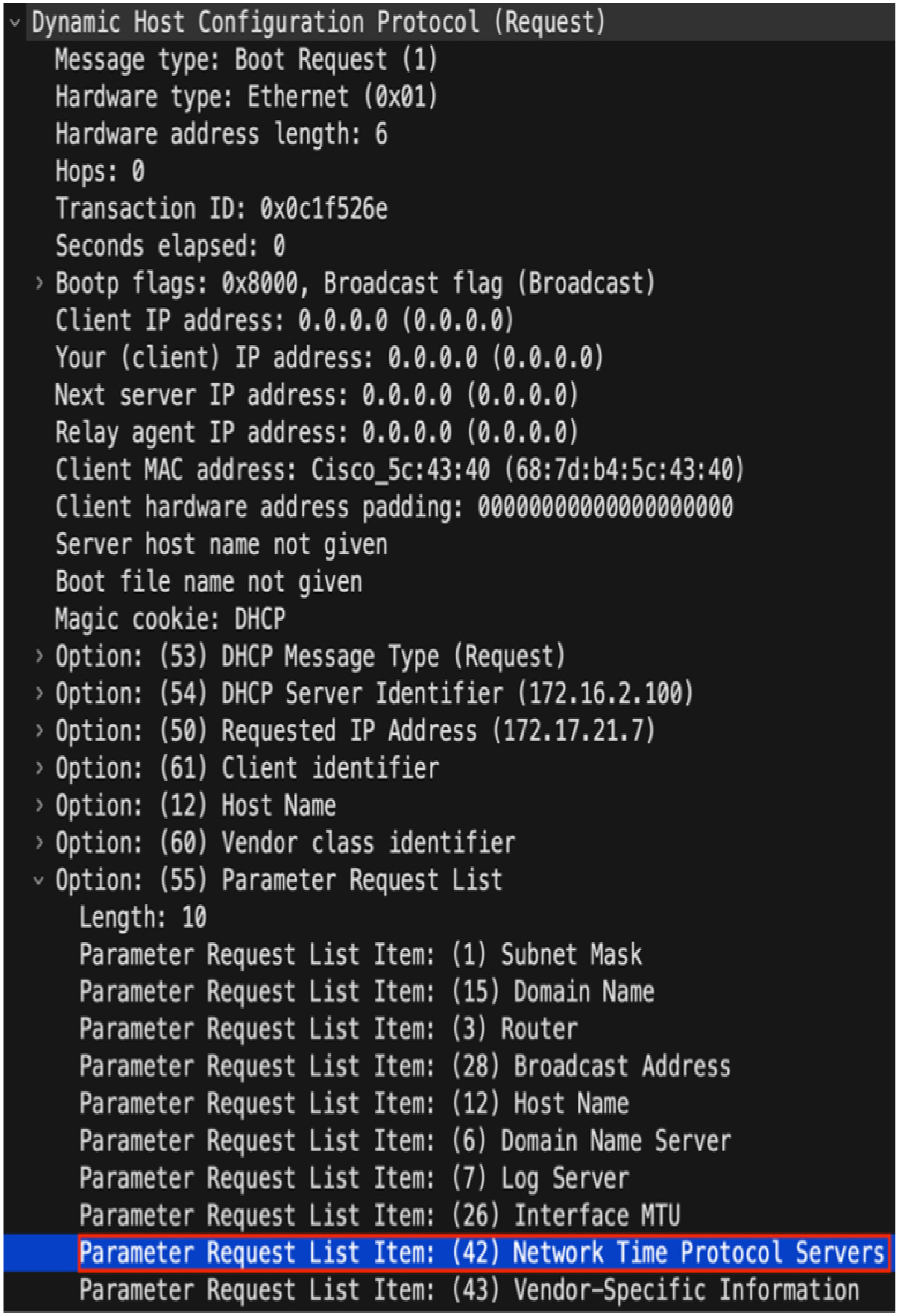
Device Name	Issue To	Certificate Serial Number	Status	Valid From	Valid To	Actions
Site4-FE.muc07cxlab.cisco.com	C9300-24P_FOC2520L6J9	4da8401c6e4ace35	✓	May 17, 2023, 10:40 AM	May 16, 2024, 10:40 AM	Revoke
Ji_Stack.muc07cxlab.cisco.com	C9300-24UX_FOC2509L5W0	1751aa6452465282	✓	May 09, 2023, 03:24 PM	May 08, 2024, 03:24 PM	Revoke
Site4-WLC.muc07cxlab.cisco.com	C9800-CL-K9_9O8N08LUEY7	4178a7b6a0fb7241	✓	Apr 21, 2023, 05:17 PM	Apr 20, 2024, 05:17 PM	Revoke

Figure 7-4 *Issued Device Certificates*

Time Management

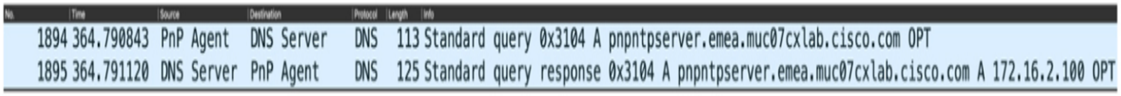
Although the subject of time management may sound unusual in this case, it is very important and relevant in the context of the Plug-and-Play implementation that the correct sanity is present to ensure the certificate validity. Depending on the platform that is performing the Plug-and-Play process, different capabilities may be used to ensure that the time on the platform is valid.

For some platforms, the DHCP Capability list that is shared during the DHCP discovery phase contains the capability to interpret DHCP Option 42 data, as detailed in Figure 7-5, thus allowing the platform to configure a DHCP-provided NTP server to perform time synchronization. On other platforms, such as the Cisco Catalyst Series switches, the pnpntpserver DNS A-Record, which allows for network nodes performing plug and play, to retrieve a mapping to the networks NTP server based on a known DNS record can be used, as seen in Figure 7-6, or alternatively when Cisco Switches are performing plug and play using Cisco Plug-and-Play Connect, the use of the DNS A-Record time-ntp.cisco.com, as in Figure 7-7, may also be utilized in order to synchronize valid time and date information.



```
⌄ Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x0c1f526e
    Seconds elapsed: 0
  › Bootp flags: 0x8000, Broadcast flag (Broadcast)
    Client IP address: 0.0.0.0 (0.0.0.0)
    Your (client) IP address: 0.0.0.0 (0.0.0.0)
    Next server IP address: 0.0.0.0 (0.0.0.0)
    Relay agent IP address: 0.0.0.0 (0.0.0.0)
    Client MAC address: Cisco_5c:43:40 (68:7d:b4:5c:43:40)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  › Option: (53) DHCP Message Type (Request)
  › Option: (54) DHCP Server Identifier (172.16.2.100)
  › Option: (50) Requested IP Address (172.17.21.7)
  › Option: (61) Client identifier
  › Option: (12) Host Name
  › Option: (60) Vendor class identifier
  ⌄ Option: (55) Parameter Request List
      Length: 10
      Parameter Request List Item: (1) Subnet Mask
      Parameter Request List Item: (15) Domain Name
      Parameter Request List Item: (3) Router
      Parameter Request List Item: (28) Broadcast Address
      Parameter Request List Item: (12) Host Name
      Parameter Request List Item: (6) Domain Name Server
      Parameter Request List Item: (7) Log Server
      Parameter Request List Item: (26) Interface MTU
      Parameter Request List Item: (42) Network Time Protocol Servers
      Parameter Request List Item: (43) Vendor-Specific Information
```

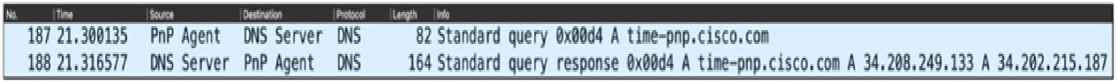
Figure 7-5 *DHCP Server Option 42 NTP*



```
No.  Time        Source      Destination  Protocol  Length  Info
1894 364.790843  PnP Agent   DNS Server   DNS       113     Standard query 0x3104 A pnpntpserver.emea.muc07cxlab.cisco.com OPT
1895 364.791120  DNS Server  PnP Agent    DNS       125     Standard query response 0x3104 A pnpntpserver.emea.muc07cxlab.cisco.com A 172.16.2.100 OPT
```

Figure 7-6 *pnpntpserver Domain Name Resolution*

DNS Server Entry time-ntp.cisco.com

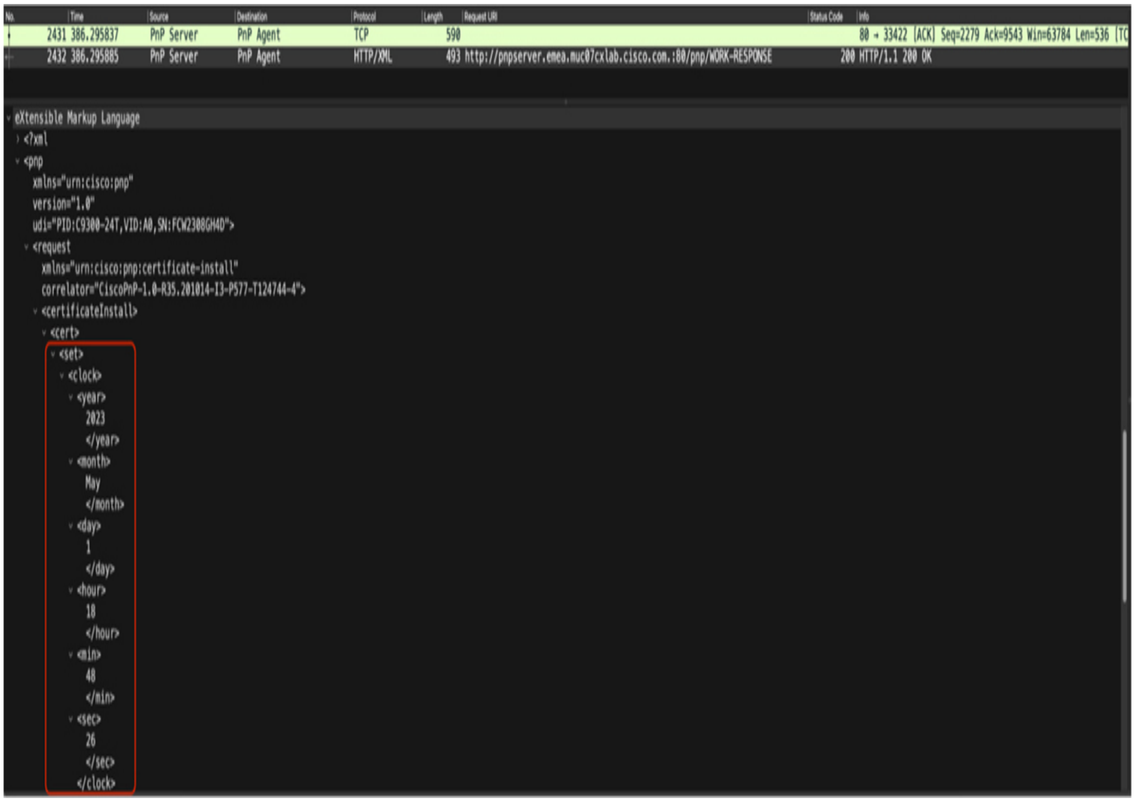


```
No. Time       Source      Destination  Protocol  Length  Info
187 21.300135  PnP Agent   DNS Server   DNS        82     Standard query 0x00d4 A time-pnp.cisco.com
188 21.316577  DNS Server  PnP Agent    DNS       164     Standard query response 0x00d4 A time-pnp.cisco.com A 34.208.249.133 A 34.202.215.187
```

Figure 7-7 *time-ntp.cisco.com Domain Name Resolution*

In addition to the methods that have been discussed to ensure a valid and up to date time synchronization state in time allocation, the Plug-and-Play process on the server side of some of the Cisco implementations may take further actions toward proactively configuring the correct time, in conjunction with the provisioning of a certificate.

Figure 7-8 shows how Catalyst Center provisions a valid time during its certificate installation phase. This action negates the need for the NTP server to be preconfigured via DNS or DHCP at the time of Plug-and-Play onboarding.



```
No.   Time         Source      Destination  Protocol  Length  Request URI                                                  Status Code  Info
2431  386.295837   PnP Server  PnP Agent    TCP       590                                                                               80 → 33422 [ACK] Seq=2279 Ack=9543 Win=63784 Len=536 [TC
2432  386.295885   PnP Server  PnP Agent    HTTP/XML  493     http://pnpserver.emea.muc07cxlab.cisco.com.:80/pnp/WORK-RESPONSE  200  HTTP/1.1 200 OK

eXtensible Markup Language
 <?xml
 <pnp
   xmlns="urn:cisco:pnp"
   version="1.0"
   udi="PID:C9300-24T,VID:A0,SN:FCW2308GH4D">
  <request
    xmlns="urn:cisco:pnp:certificate-install"
    correlator="CiscoPnP-1.0-R35.201014-I3-P577-T124744-4">
   <certificateInstall>
    <cert>
     <set>
      <clock>
       <year>
        2023
        </year>
       <month>
        May
        </month>
       <day>
        1
        </day>
       <hour>
        18
        </hour>
       <min>
        48
        </min>
       <sec>
        26
        </sec>
       </clock>
```

Figure 7-8 *Time Provisioning in Combination with Certificate Transmission*

Using IPv4 DHCP to Perform Plug-and-Play

By far, the most common method to onboard Cisco devices while using solutions such as Cisco Catalyst Center, which evolved in enterprise networks from the Cisco APIC-EM solution, is to use IPv4 DHCP to perform Plug-and-Play. In service provider and some defense sector domains, Network Services Orchestrator uses DHCP in conjunction with specific bootstrap options for zero touch or Plug-and-Play provisioning. Historically, Option 43 was used in Cisco wireless networks as a means for lightweight access points (APs) to identify the wireless LAN controllers (WLCs) for which they would generate a control and data connection using the CAPWAP protocol.

Upon the release of Plug-and-Play, using DHCP Option 43 became a further means by which a factory default network device has a means to locate the IP address or, in certain circumstances, the FQDN of its corresponding Plug-and-Play server.

Before we delve into all the different options associated with formatting DHCP Option 43 addressing, let's examine how the DHCP server that is responsible for providing the router, switch, wireless controller, or AP with its respective IP address is configured.

DHCP Server Scope

The service IP address scope represents the range of IP addresses that are allocated to a specific subnet. Let's consider the example of a virtual LAN (VLAN) that is dedicated for access points. These access points would be planned with an addressing allocation—for example, 192.168.129.0/24. In this context, we would refer to the range of usable addresses to allocate to the client devices as the *scope*. Most scopes also include exclusions, to avoid existing infrastructure addresses, such as the default gateway IP address, from being provided, and resulting in duplicate addressing allocations.

In modern DHCP servers, further exclusion validation checks, such as probing addresses in use on the network, may also take place automatically.

As part of the scope allocation (RFC 2132), attributes that are provided to dynamically provisioned hosts are

- IP Address
- Subnet Mask: Option Number 1
- Default Gateway: Option Number 3

Note

While alternatives to deploying the default gateway may exist, such as using DHCP Options 33 or 121 or 249, the ability of network platforms to consume and use this addressing during the Plug-and-Play phase may vary on a platform-by-platform basis.

DNS Server: DHCP Option 6

The addition of the DNS server attribute is important when using DHCP Option 43 in conjunction with a fully qualified domain name, rather than when using an IPv4 address. Failure to provide a reachable DNS server will result in the PnP process failing to identify the IP address of the server that it needs to communicate with.

Domain Name: DHCP Option 15

The domain name option can be used to provide an existing context to which default level in the DNS hierarchy a device may be located—for example, apac.acme.corp versus amer.acme.corp.

Vendor Class Identifier: DHCP Option 60

While the options that we have described so far indicate key information that is used, such as the client that is receiving the IP address from the DHCP server, there is an important option that is commonly used to provide the server with information about the DHCP client device itself. Within a DHCP scope, there is the possibility that different types of systems may exist: systems that have already been provisioned and have concluded their Plug-and-Play onboarding, systems that may need to directly discover control servers such as IP phones or other equipment that may not react well to the provisioning of DHCP options, such as Option 43, when they are not needed. In such scenarios, the DHCP Option 60 Vendor Class Identifier field can be used. This field is commonly used within a scoped IP address range to allocate devices that match a specific string (hex or ASCII) into subscopes that represent a portion of the allocated range or subscopes that represent the provisioning of different options.

Unlike DHCP Option 43, which was described earlier, Option 60 is transmitted in the DHCP discover packet to share information upstream, as shown in Figure 7-9.

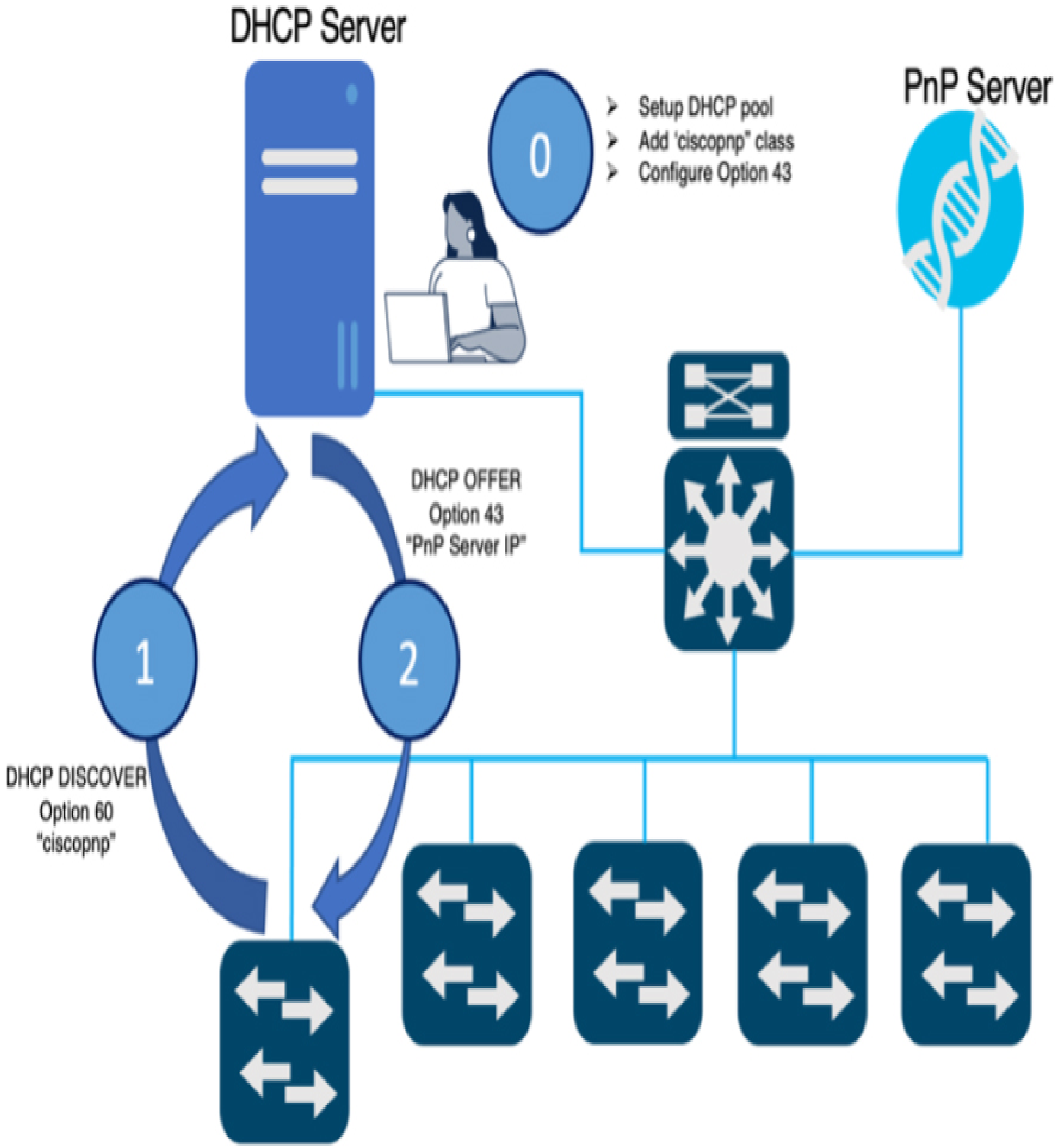
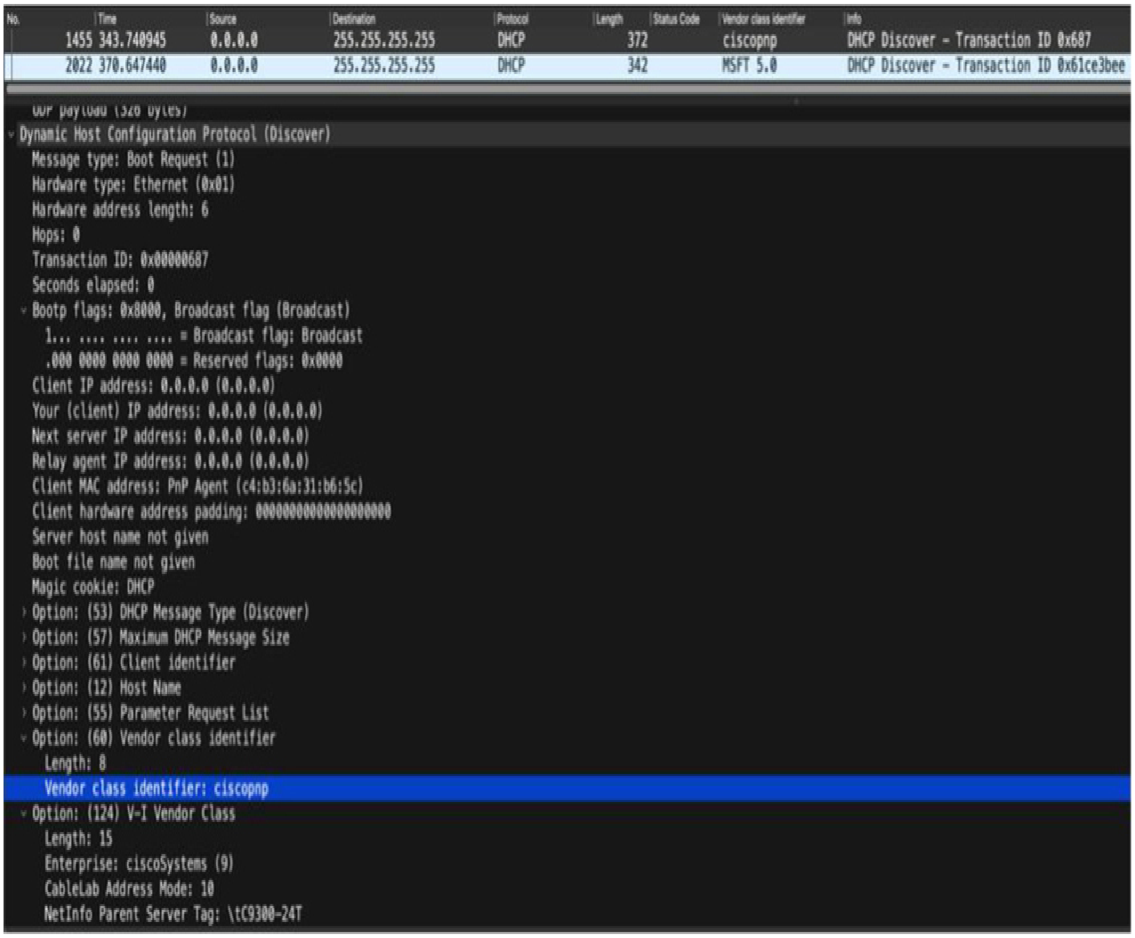


Figure 7-9 *DHCP Option 60 Flow in PnP*

A simple example of how this may be used comes from the world of wireless LAN deployments. In the past, customers with access point models used older generations of software and hardware that could not utilize the newer wireless LAN controller series. For these devices, it was important that they continue to communicate with their legacy controller to ensure the right interoperability was available. This included Return Material Authorization (RMA) or replacement legacy APs from the stockroom that may need to onboard via DHCP Option 43 and be directed toward the legacy controller. In contrast, for newer APs, the requirement would be to communicate with a newer wireless LAN controller and receive its IP address via Option 43. This scenario would have been simple to achieve if the devices were in two totally different subnets, using dedicated IP scopes with fixed Option 43 mappings to the old and new controllers. However, if the devices are in the same IP subnet and range, then things become more complicated. As such, the use of Option 60 is important to match on the string for the old and/or new vendor class identifiers that can support the identification of the vendor and under certain circumstances a product name in the DHCP packet (VCI strings) to provide the right DHCP scope.

> **Note**
>
> VCI strings on Cisco devices change from the default value of ciscopnp, as shown in Figure 7-10, after the Plug-and-Play process has concluded. Typically, the change represents something that would allow the identification of the device model, such as C9136 representing a Cisco 9136 access point.



```
No.   Time         Source    Destination      Protocol  Length  Status Code  Vendor class identifier  Info
1455  343.740945   0.0.0.0   255.255.255.255  DHCP      372                  ciscopnp                 DHCP Discover - Transaction ID 0x687
2022  370.647440   0.0.0.0   255.255.255.255  DHCP      342                  MSFT 5.0                 DHCP Discover - Transaction ID 0x61ce3bee

   UDP payload (320 bytes)
 Dynamic Host Configuration Protocol (Discover)
   Message type: Boot Request (1)
   Hardware type: Ethernet (0x01)
   Hardware address length: 6
   Hops: 0
   Transaction ID: 0x00000687
   Seconds elapsed: 0
   Bootp flags: 0x8000, Broadcast flag (Broadcast)
     1... .... .... .... = Broadcast flag: Broadcast
     .000 0000 0000 0000 = Reserved flags: 0x0000
   Client IP address: 0.0.0.0 (0.0.0.0)
   Your (client) IP address: 0.0.0.0 (0.0.0.0)
   Next server IP address: 0.0.0.0 (0.0.0.0)
   Relay agent IP address: 0.0.0.0 (0.0.0.0)
   Client MAC address: PnP Agent (c4:b3:6a:31:b6:5c)
   Client hardware address padding: 00000000000000000000
   Server host name not given
   Boot file name not given
   Magic cookie: DHCP
   Option: (53) DHCP Message Type (Discover)
   Option: (57) Maximum DHCP Message Size
   Option: (61) Client identifier
   Option: (12) Host Name
   Option: (55) Parameter Request List
   Option: (60) Vendor class identifier
     Length: 8
     Vendor class identifier: ciscopnp
   Option: (124) V-I Vendor Class
     Length: 15
     Enterprise: ciscoSystems (9)
     CableLab Address Mode: 10
     NetInfo Parent Server Tag: \tC9300-24T
```

Figure 7-10 *DHCP Discover with Default ciscopnp VCI Value*

DHCP Option 43

Looking at the options for using Option 43 with PnP, at first, can be quite confusing. This is especially true for individuals who have a background working with wireless, where most commonly the strings used for wireless LAN controller discovery are limited to a short hexadecimal string representing the IP address and the number of WLCs to use.

When you're interfacing directly with a Plug-and-Play server, the formatting provides a lot more control and power in terms of what is being used for the communications. Looking at Example 7-2, which was taken from a Cisco Switch, you can see an ASCII string that includes numeric and hexadecimal values.

Example 7-2 *IP Pool with ASCII DHCP Option 43 Syntax*

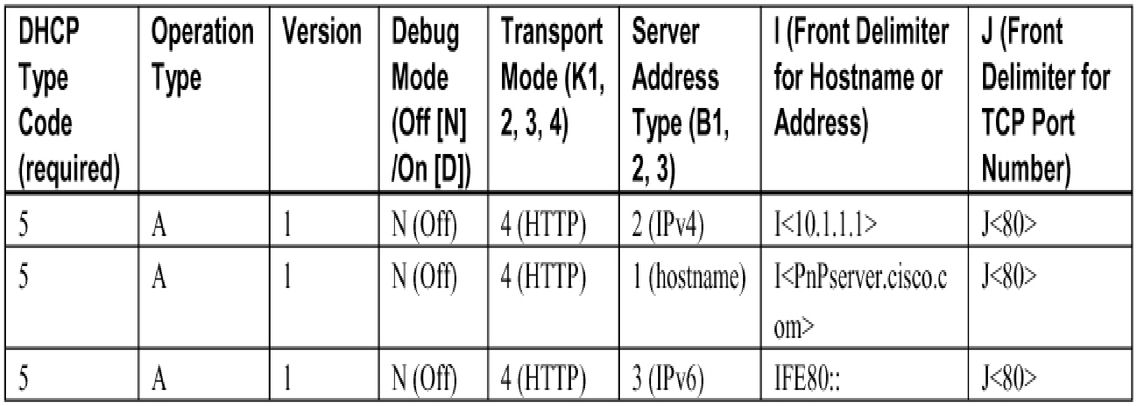
```
ip dhcp pool PnP_Pool
network 10.20.30.0 255.255.255.0
default-router 10.20.30.1
option 43 ascii "5A1D;B2;K4;I10.1.1.1;J80;"
domain-name zerotrustbook.cisco.com
dns-server 10.19.80.4
```

Note

Depending on the DHCP server implementation, hexadecimal could be preferred instead of ASCII.

In Table 7-1, the DHCP Option 43 content translates to specific functions. Let’s break down the options listed within that field.

Table 7-1 *DHCP Option 43 PnP Definitions*



DHCP Type Code (required)	Operation Type	Version	Debug Mode (Off [N] /On [D])	Transport Mode (K1, 2, 3, 4)	Server Address Type (B1, 2, 3)	I (Front Delimiter for Hostname or Address)	J (Front Delimiter for TCP Port Number)
5	A	1	N (Off)	4 (HTTP)	2 (IPv4)	I<10.1.1.1>	J<80>
5	A	1	N (Off)	4 (HTTP)	1 (hostname)	I<PnPserver.cisco.com>	J<80>
5	A	1	N (Off)	4 (HTTP)	3 (IPv6)	IFE80::	J<80>

Using DNS to Perform Plug-and-Play

Whereas DHCP tends to remain a common option for use within enterprise customer networks using Cisco IOS XE products and portfolios, for customers in the service provider space or Cisco Meraki customers, using DNS is much more common. A key advantage with the use of DNS is independence from the needs of special DHCP configurations for Plug-and-Play devices to be onboarded.

In the context of the Catalyst Center platform, operators may opt toward using certificates that are based exclusively on FQDNs. This approach has become quite popular in customer accounts where restrictive certificate authorities are in use. When the FQDN-based approach is used to perform PnP for devices, a default A-record convention is required. This specific A-record is the host entry PnPserver. When a network device's Plug-and-Play agent is searching for a viable onboarding host via the FQDN method, it will specifically search for PnPserver within its existing domain. If no domain is provided, it will attempt to identify a PnPserver entry at the same subdomain where the system is currently provisioned or has configured itself within upon receiving an IP address using a reverse DNS lookup.

This approach brings several advantages over the Option 43 onboarding method that we covered previously. While the Option 43 method does have numerous capable options, using the FQDN method potentially allows a more simplistic means for the operator to allocate global PnP servers to them to be onboarded devices.

Let's look at a scenario where we have a global enterprise customer as in Figure 7-11. This customer has three Catalyst Center clusters that are deployed across the globe. One is in the Asia Pacific, one in Europe, and one in the Americas, using three different subdomains using IPv4 DHCP Domain Name (option 15). These subdomain allocations can be applied to the respective PnP provisioning VLANs in the respective geographies.

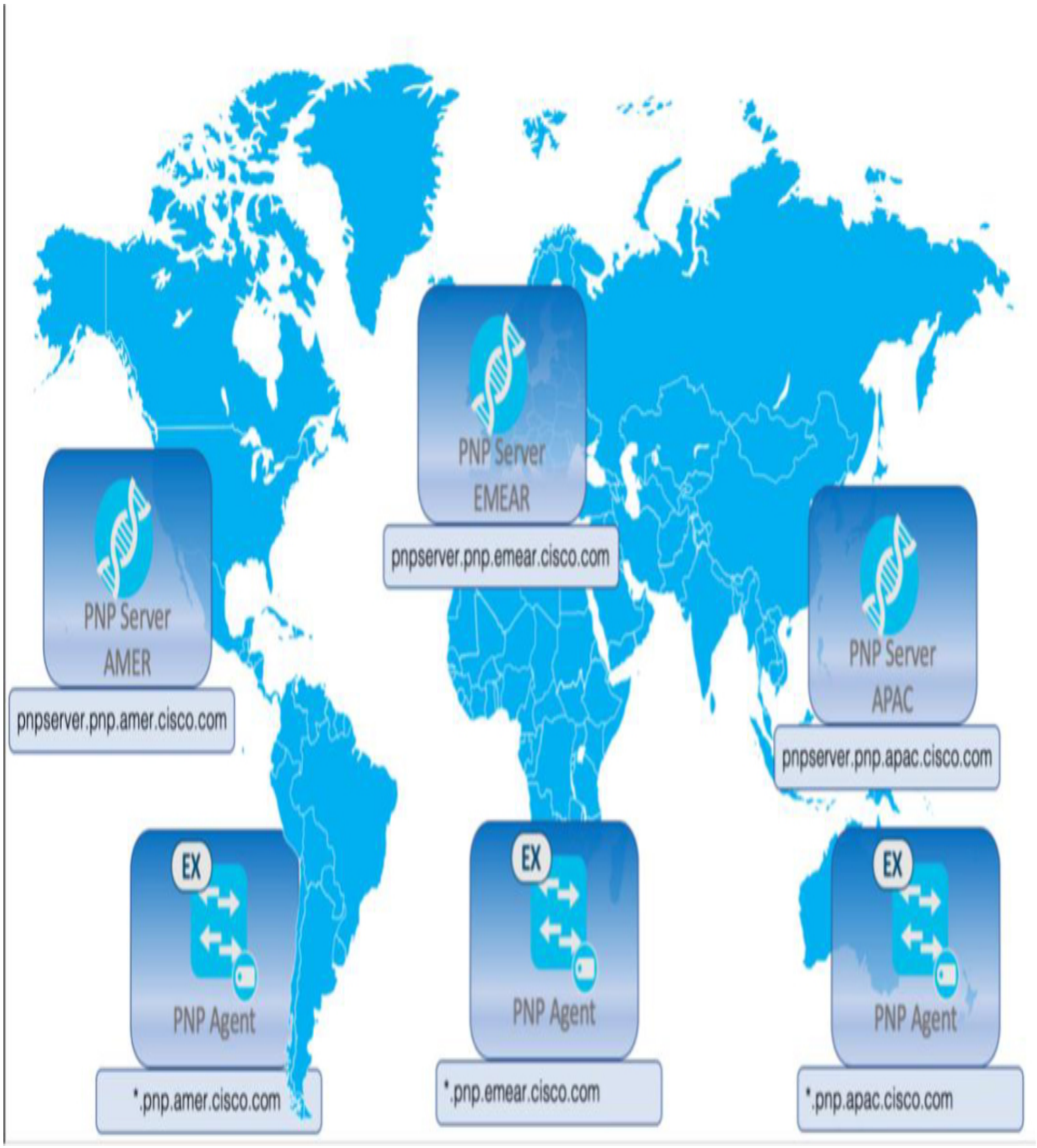


Figure 7-11 *Global PnP-Server Deployment*

Plug-and-Play Connect

The Plug-and-Play Connect feature can be used by supporting devices that want to participate in the Plug-and-Play process and may not have localized mapping to a Plug-and-Play server that exists within the local enterprise network.

For PnP Connect to function properly, DNS resolution of the pnpconnect.cisco.com URL is required, as is the ability to route to the respective IP address that the FQDN resolves to. Since connectivity to the portal commonly traverses out of a private corporate network, requisite firewall rules also need to be opened to support the communications from the network device to the Cisco Plug-and-Play Connect cloud-based discovery portal. The on-boarding flow for PnP Connect is illustrated in more granularity in Figure 7-12.

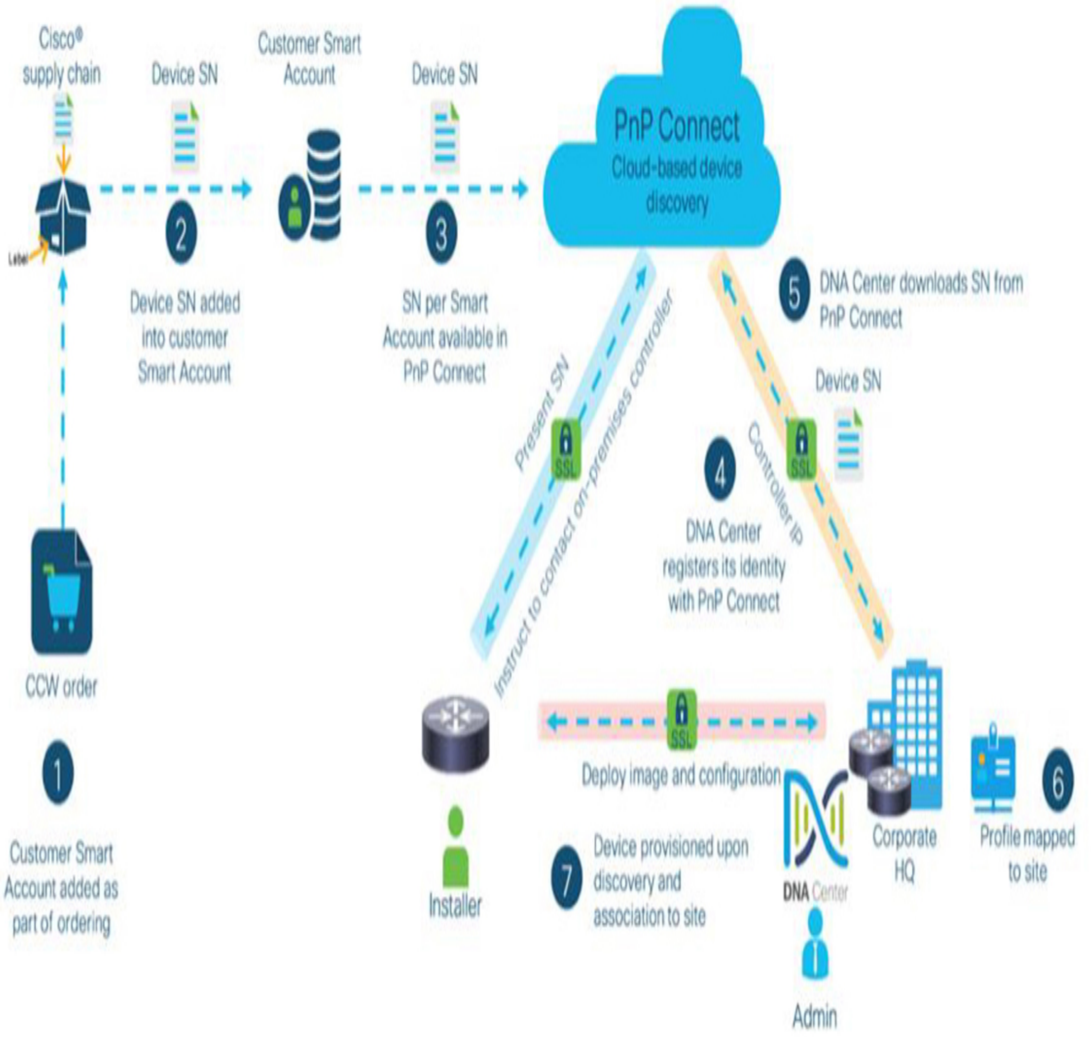

Cisco® supply chain
Label
Device SN
Customer Smart Account
Device SN
PnP Connect
Cloud-based device discovery
2
Device SN added into customer Smart Account
3
SN per Smart Account available in PnP Connect
5
DNA Center downloads SN from PnP Connect
Device SN
Present SN
Instruct to contact on-premises controller
SSL
4
DNA Center registers its identity with PnP Connect
Controller IP
SSL
CCW order
1
Customer Smart Account added as part of ordering
SSL
Deploy image and configuration
Installer
7
Device provisioned upon discovery and association to site
DNA Center
Corporate HQ
Profile mapped to site
6
Admin

Figure 7-12 *PnP Connect Workflow*

Startup VLAN

When you’re booting up factory-new Cisco switches and routers, all physical interfaces reside in VLAN 1. This information is helpful for several reasons. During bootup within a Layer 2 domain, key protocols such as DTP use this VLAN to allow for negotiations. Furthermore, this VLAN is quite commonly the native VLAN that is used in peer devices when no extra configurations are applied.

With that said, in many scenarios, the default VLAN that is desired for use within the network may not be VLAN 1, particularly in scenarios where networks may be using things such as macrosegmentation (VRFs) or handoffs to Carrier Edge routers from an SD-WAN or SP domain or domains that have disabled VLAN 1 for security reasons.

In such scenarios, it is possible to change the default VLAN for the PnP process on the adjacent device to the network device running the PnP agent. In this case, the following command can be configured:

```
pnp startup vlan <vlan id>
```

The configuration does not have any further effect on the device where it was configured.

LACP Usage with PnP

In certain scenarios, you might need to configure Plug-and-Play to interface with upstream devices that may need to generate a port channel, such as an upstream distribution switch. For such a scenario, you can use the **no port-channel standalone-disable** command within a deployment template similar to the one shown in Example 7-3.

Example 7-3 *Preparing Upstream Devices for Plug-and-Play with Port Channels*

```
!
Vlan 192
 name MGMT-VLAN
```

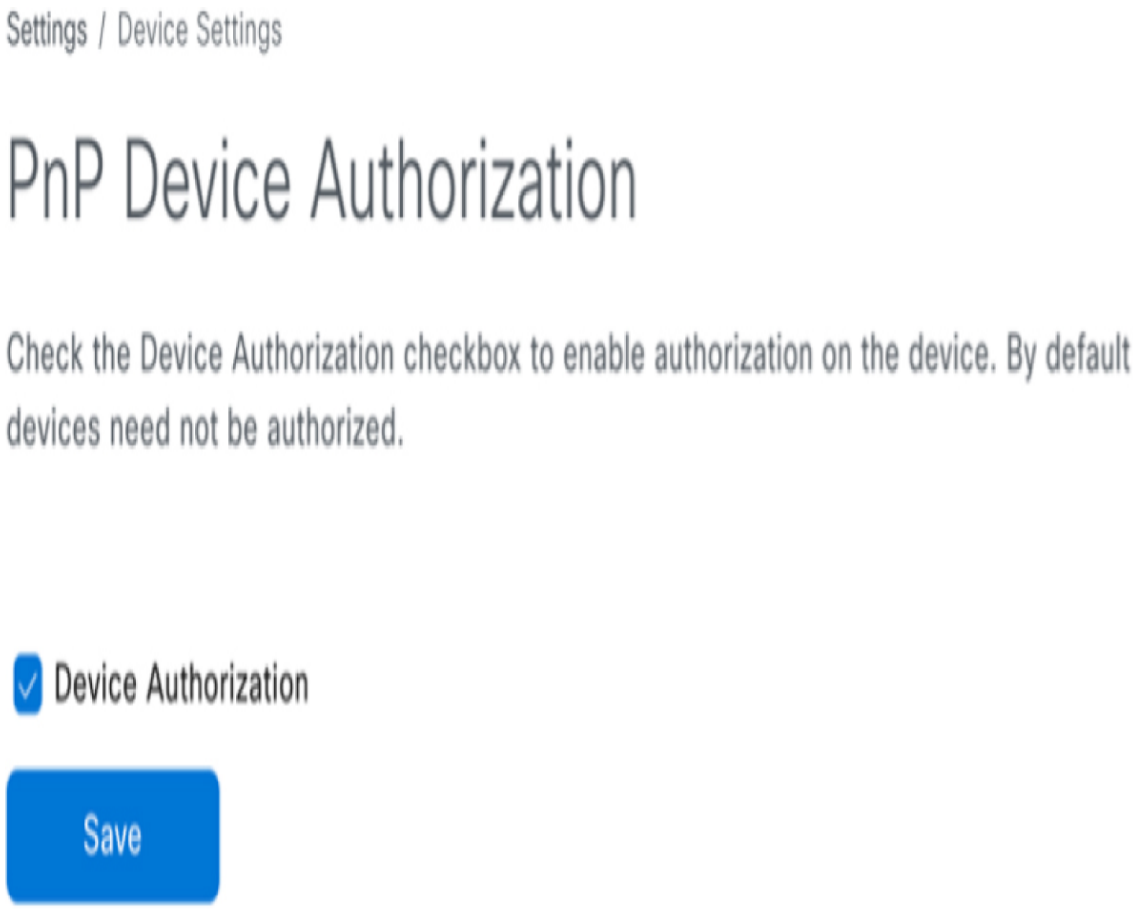
```
!
Interface vlan 192
 ip address dhcp
 no shut
!
Interface range ten 1/0/1, ten 2/0/1
 switchport mode trunk
 switchport trunk native vlan 192
 channel-group 99 mode active
!
Interface port-channel 99
 no port-channel standalone-disable
!
PnP profile PnP-zero-touch
 transport http ipv4 {{DNAC-IP}} source vlan192
exit
```

Authorization of PnP Devices

You can control access to a network medium in the context of Plug-and-Play in several ways, depending on the deployment model that is in use on the Plug-and-Play server. Some Plug-and-Play functionality requires explicit claiming to take place, using the serial number of the device prior to it being provisioned with a day zero configuration.

Other network functions, such as the deployment of a supplicant-based extended node switch in the SD-Access network, if left with the default configurations, will automatically provision, extending the network perimeter. Although this is a desirable behavior for some customers, other customers may perceive the dynamic growth of the network perimeter to represent a security concern and still would like to retain control over which devices can be onboarded via the Plug-and-Play process.

The principles of zero trust can be aligned within Catalyst Center to ensure that devices are explicitly authorized by the operator prior to any provisioning of day zero configuration, software, and licensing. This can be configured within **System Settings > Device Settings**, as shown in Figure 7-13.



Settings / Device Settings

PnP Device Authorization

Check the Device Authorization checkbox to enable authorization on the device. By default devices need not be authorized.

☑ Device Authorization

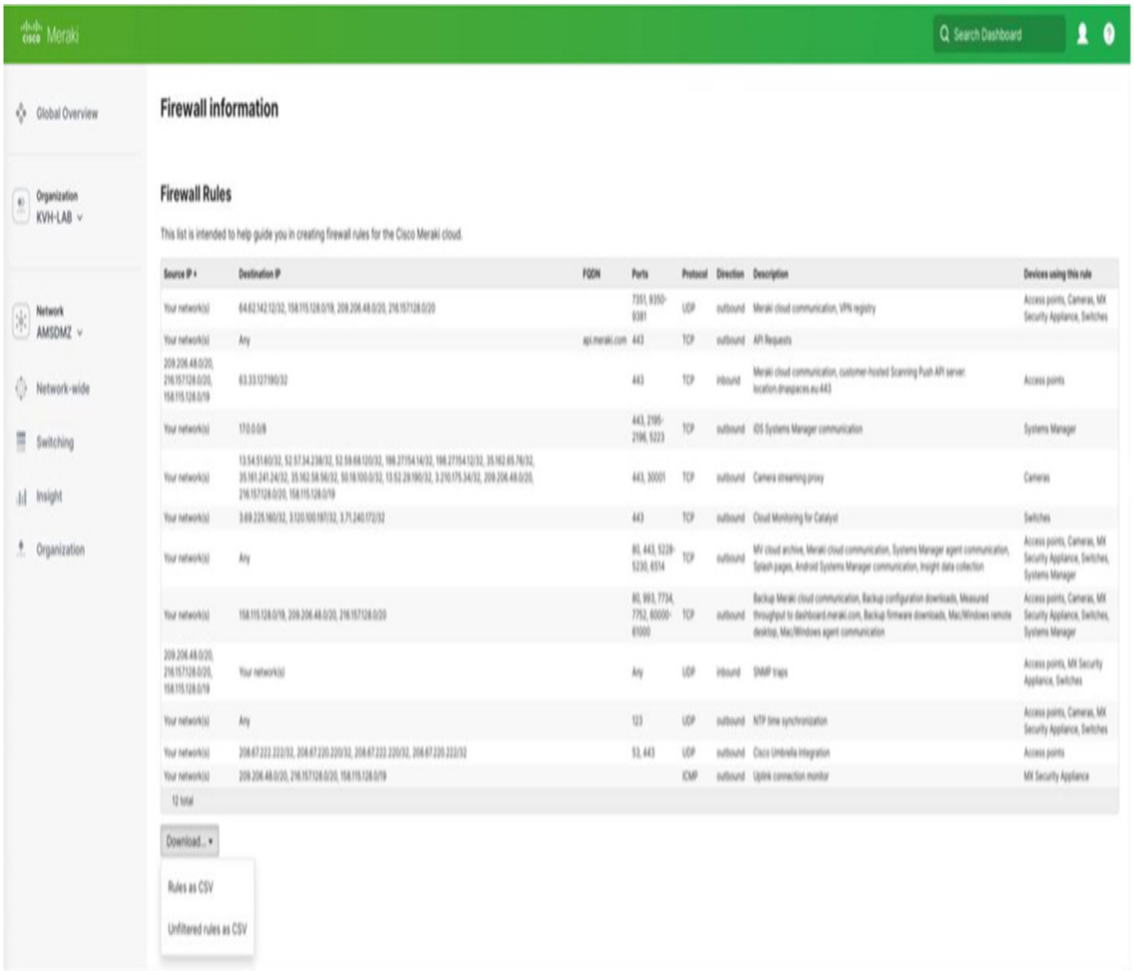
Figure 7-13 *PnP Security Feature to Permit Only Authorized Devices to Be Provisioned*

Note

Authorization is an optional step to allow for a higher level of security within customer environments. This option impedes devices from being claimed into the network, prior to their existence being accepted.

Meraki Onboarding Flow

Onboarding a Meraki device to the dashboard requires access to the Internet to allow the onboarding device to resolve the key and critical URLs associated with the organization in its respective region. For devices that may have a firewall in the middle, a key overview of the ports that need to be opened per service is available, as depicted in Figure 7-14. An updated overview of these details can be derived from the **Firewall** ruleset tab at the top right corner of the dashboard.



Firewall information

Firewall Rules

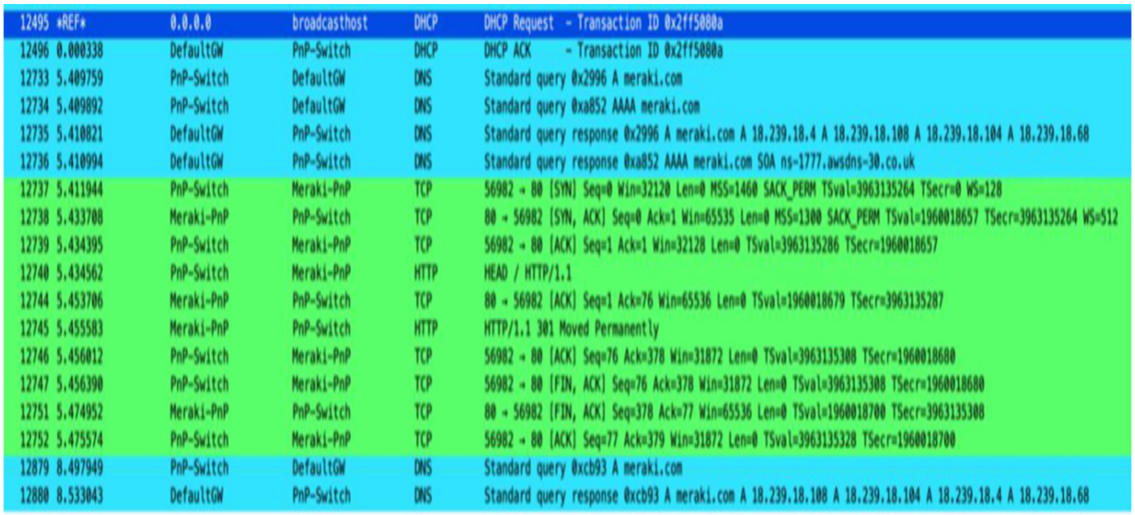
This list is intended to help guide you in creating firewall rules for the Cisco Meraki cloud.

Source IP	Destination IP	FQDN	Ports	Protocol	Direction	Description	Devices using this rule
Your network(s)	64.62.142.12/32, 158.115.128.0/19, 209.206.48.0/20, 216.157.128.0/20		7351, 9350-9381	UDP	outbound	Meraki cloud communication, VPN registry	Access points, Cameras, MX Security Appliance, Switches
Your network(s)	Any	api.meraki.com	443	TCP	outbound	API Requests	
209.206.48.0/20, 216.157.128.0/20, 158.115.128.0/19	63.33.127.190/32		443	TCP	inbound	Meraki cloud communication, customer-hosted Scanning Push API server. location.dnaspaces.eu:443	Access points
Your network(s)	17.0.0.0/8		443, 2195-2196, 5223	TCP	outbound	iOS Systems Manager communication	Systems Manager
Your network(s)	13.54.51.60/32, 52.57.34.238/32, 52.59.68.120/32, 198.27.154.14/32, 198.27.154.12/32, 35.162.65.76/32, 35.161.241.24/32, 35.162.58.56/32, 50.18.100.0/32, 13.52.29.190/32, 3.210.175.34/32, 209.206.48.0/20, 216.157.128.0/20, 158.115.128.0/19		443, 3001	TCP	outbound	Camera streaming proxy	Cameras
Your network(s)	3.69.225.160/32, 3.120.100.197/32, 3.71.240.172/32		443	TCP	outbound	Cloud Monitoring for Catalyst	Switches
Your network(s)	Any		80, 443, 5228-5230, 6514	TCP	outbound	MV cloud archive, Meraki cloud communication, Systems Manager agent communication, Splash pages, Android Systems Manager communication, Insight data collection	Access points, Cameras, MX Security Appliance, Switches, Systems Manager
Your network(s)	158.115.128.0/19, 209.206.48.0/20, 216.157.128.0/20		80, 993, 7734, 7752, 60000-61000	TCP	outbound	Backup Meraki cloud communication, Backup configuration downloads, Measured throughput to dashboard.meraki.com, Backup firmware downloads, Mac/Windows remote desktop, Mac/Windows agent communication	Access points, Cameras, MX Security Appliance, Switches, Systems Manager
209.206.48.0/20, 216.157.128.0/20, 158.115.128.0/19	Your network(s)		Any	UDP	inbound	SNMP traps	Access points, MX Security Appliance, Switches
Your network(s)	Any		123	UDP	outbound	NTP time synchronization	Access points, Cameras, MX Security Appliance, Switches
Your network(s)	208.67.222.222/32, 208.67.220.220/32, 208.67.222.220/32, 208.67.220.222/32		53, 443	UDP	outbound	Cisco Umbrella Integration	Access points
Your network(s)	209.206.48.0/20, 216.157.128.0/20, 158.115.128.0/19			ICMP	outbound	Uplink connection monitor	MX Security Appliance

12 total

Figure 7-14 *Meraki Firewall Ruleset Requirements*

To get a more detailed perspective of the flow of traffic involved in onboarding a new Meraki device, the packet flow in Figure 7-15 provides further perspective on how the communications are established.



```
12495 *REF*     0.0.0.0     broadcasthost DHCP DHCP Request  - Transaction ID 0x2ff5080a
12496 0.000338  DefaultGW   PnP-Switch    DHCP DHCP ACK      - Transaction ID 0x2ff5080a
12733 5.409759  PnP-Switch  DefaultGW     DNS  Standard query 0x2996 A meraki.com
12734 5.409892  PnP-Switch  DefaultGW     DNS  Standard query 0xa852 AAAA meraki.com
12735 5.410821  DefaultGW   PnP-Switch    DNS  Standard query response 0x2996 A meraki.com A 18.239.18.4 A 18.239.18.108 A 18.239.18.104 A 18.239.18.68
12736 5.410994  DefaultGW   PnP-Switch    DNS  Standard query response 0xa852 AAAA meraki.com SOA ns-1777.awsdns-30.co.uk
12737 5.411944  PnP-Switch  Meraki-PnP    TCP  56982 → 80 [SYN] Seq=0 Win=32120 Len=0 MSS=1460 SACK_PERM TSval=3963135264 TSecr=0 WS=128
12738 5.433708  Meraki-PnP  PnP-Switch    TCP  80 → 56982 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 SACK_PERM TSval=1960018657 TSecr=3963135264 WS=512
12739 5.434395  PnP-Switch  Meraki-PnP    TCP  56982 → 80 [ACK] Seq=1 Ack=1 Win=32128 Len=0 TSval=3963135286 TSecr=1960018657
12740 5.434562  PnP-Switch  Meraki-PnP    HTTP HEAD / HTTP/1.1
12744 5.453706  Meraki-PnP  PnP-Switch    TCP  80 → 56982 [ACK] Seq=1 Ack=76 Win=65536 Len=0 TSval=1960018679 TSecr=3963135287
12745 5.455583  Meraki-PnP  PnP-Switch    HTTP HTTP/1.1 301 Moved Permanently
12746 5.456012  PnP-Switch  Meraki-PnP    TCP  56982 → 80 [ACK] Seq=76 Ack=378 Win=31872 Len=0 TSval=3963135308 TSecr=1960018680
12747 5.456390  PnP-Switch  Meraki-PnP    TCP  56982 → 80 [FIN, ACK] Seq=76 Ack=378 Win=31872 Len=0 TSval=3963135308 TSecr=1960018680
12751 5.474952  Meraki-PnP  PnP-Switch    TCP  80 → 56982 [FIN, ACK] Seq=378 Ack=77 Win=65536 Len=0 TSval=1960018700 TSecr=3963135308
12752 5.475574  PnP-Switch  Meraki-PnP    TCP  56982 → 80 [ACK] Seq=77 Ack=379 Win=31872 Len=0 TSval=3963135328 TSecr=1960018700
12879 8.497949  PnP-Switch  DefaultGW     DNS  Standard query 0xcb93 A meraki.com
12880 8.533043  DefaultGW   PnP-Switch    DNS  Standard query response 0xcb93 A meraki.com A 18.239.18.108 A 18.239.18.104 A 18.239.18.4 A 18.239.18.68
```

Figure 7-15 *Traffic Flow—Meraki Onboarding*

Zero Touch Provisioning

The origins of Plug-and-Play provisioning initially were simplistic, providing a basic ability to load configuration files onto a device during a factory default bootup. This capability has since been extended and expanded to allow more secure methods of updates to take place, differential deployment based on platform characteristics, and chained actions such as software upgrades during the provisioning of a new device.

Provisioning activities can also extend to the deployment of licensing and pre-emptive execution, whereby existing serial numbers are already mapped to the future target provisioning actions that should take place upon the device coming online.

Foundation Configurations

Software version harmonization is a key cornerstone of network standardization, known bug and security mitigation, and stability using tried-and-tested versions of software that were validated as a complete and whole solution (solution testing) between disparate hardware models.

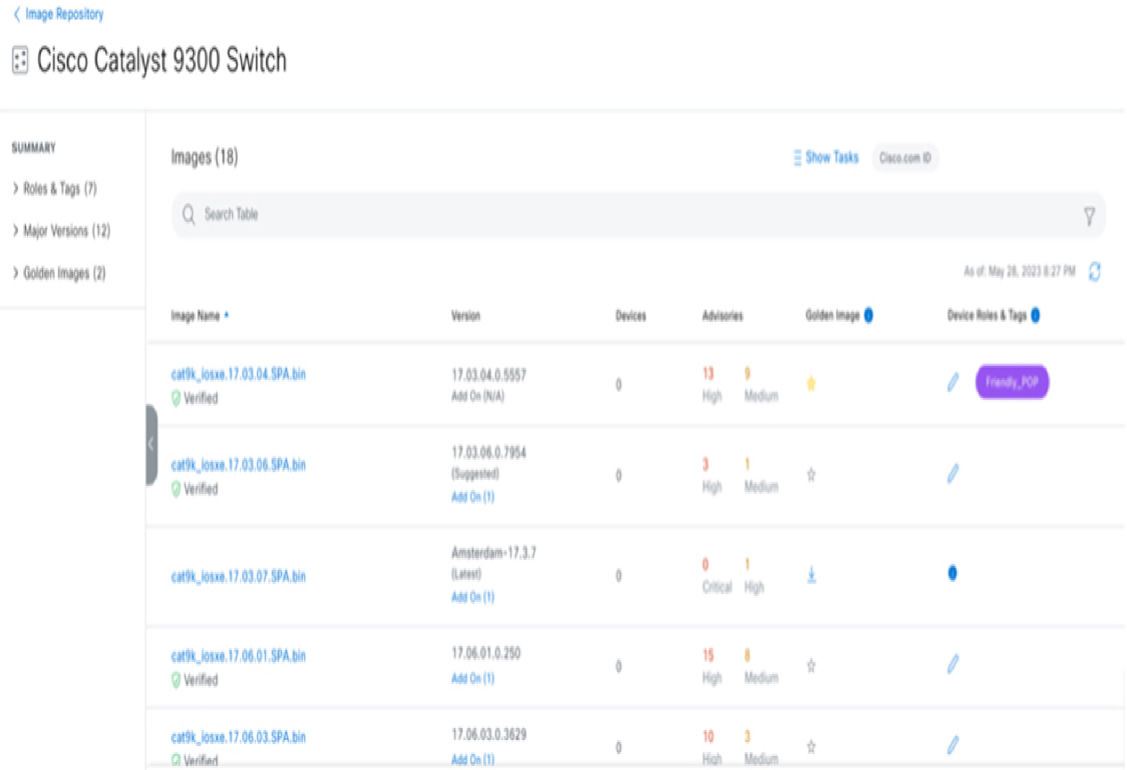
Historically, software versioning on new equipment was often a task that was reserved for customers, partners as part of dead-on-arrival (DoA) checks. During such prechecks, the installer would boot up a new device, see that it powered on, and confirm some basic command function as expected, such as **show version** or **show inventory**, followed by bringing the router, switch, or wireless controller to the target software version.

The challenge that many customers faced with this activity was that logistics in a project do not always match up with the project plan. Changes happen, deliveries are delayed, and planned resources to do the installation may become unavailable. These hurdles unfortunately can result in the planned target software versioning installed on the target system that was presumed "bug free" (if such a thing exists) no longer being in such a state, so an upgrade will be needed upon installation.

With zero touch provisioning, the onboarding process, which involves the claiming of a device and the provisioning of a certificate and a base configuration, can also load a new software image and the target licensing to ensure that its model matches up with the rest of the network standards upon being installed.

Software and Hardware Deployment Selection in Catalyst Center

Key configuration capabilities are present within Catalyst Center, thus allowing targeted software selection based on the use of platform type and role or variable-based assignment with the use of "tags," as shown in Figure 7-16, allowing for extensive flexibility when selecting software for deployment during the Plug-and-Play process. Through this granularity, differential software versioning can be tailored easily to the environment, partition, or deployment construct that is relevant to the specific environment.



< Image Repository

Cisco Catalyst 9300 Switch

SUMMARY

- Roles & Tags (7)
- Major Versions (12)
- Golden Images (2)

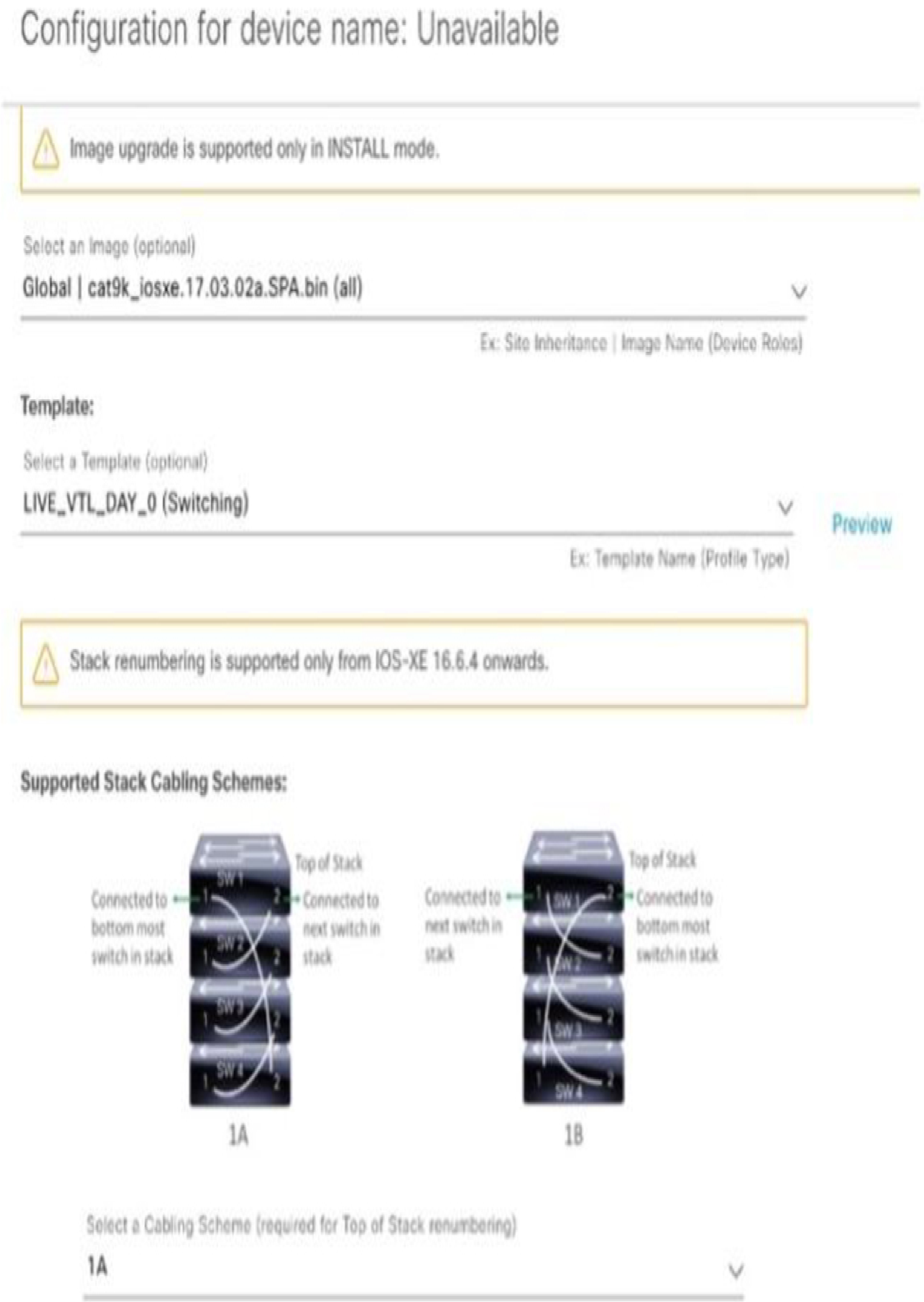
Images (18)

As of: May 28, 2023 8:27 PM

Image Name	Version	Devices	Advisories	Golden Image	Device Roles & Tags
cat9k_iosxe.17.03.04.SPA.bin Verified	17.03.04.0.5557 Add On (N/A)	0	13 High 9 Medium	★	✎ Friendly_POP
cat9k_iosxe.17.03.06.SPA.bin Verified	17.03.06.0.7954 (Suggested) Add On (1)	0	3 High 1 Medium	☆	✎
cat9k_iosxe.17.03.07.SPA.bin	Amsterdam-17.3.7 (Latest) Add On (1)	0	0 Critical 1 High	⤓	●
cat9k_iosxe.17.06.01.SPA.bin Verified	17.06.01.0.250 Add On (1)	0	15 High 8 Medium	☆	✎
cat9k_iosxe.17.06.03.SPA.bin Verified	17.06.03.0.3629 Add On (1)	0	10 High 3 Medium	☆	✎

Figure 7-16 *Tag-Based Software Selection*

In addition to the software versioning selection, for stacked devices, stack-cabling constructs, like those shown in Figure 7-17, can be selected to ensure that newly provisioned stacks can also be provisioned easily, with the right versioning, licensing, and configurations.



Configuration for device name: Unavailable

⚠ Image upgrade is supported only in INSTALL mode.

Select an Image (optional)

Global | cat9k_iosxe.17.03.02a.SPA.bin (all) ∨

Ex: Site Inheritance | Image Name (Device Roles)

Template:

Select a Template (optional)

LIVE_VTL_DAY_0 (Switching) ∨ Preview

Ex: Template Name (Profile Type)

⚠ Stack renumbering is supported only from IOS-XE 16.6.4 onwards.

Supported Stack Cabling Schemes:

Top of Stack
Connected to bottom most switch in stack
Connected to next switch in stack
SW 1
SW 2
SW 3
SW 4
1
2
1A
Top of Stack
Connected to next switch in stack
Connected to bottom most switch in stack
SW 1
SW 2
SW 3
SW 4
1B

Select a Cabling Scheme (required for Top of Stack renumbering)

1A ∨

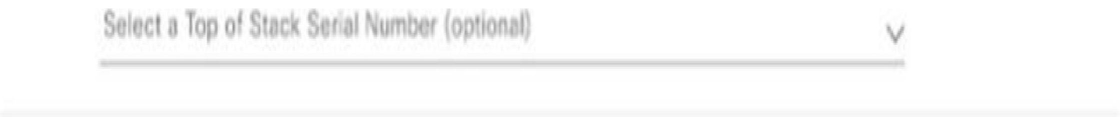
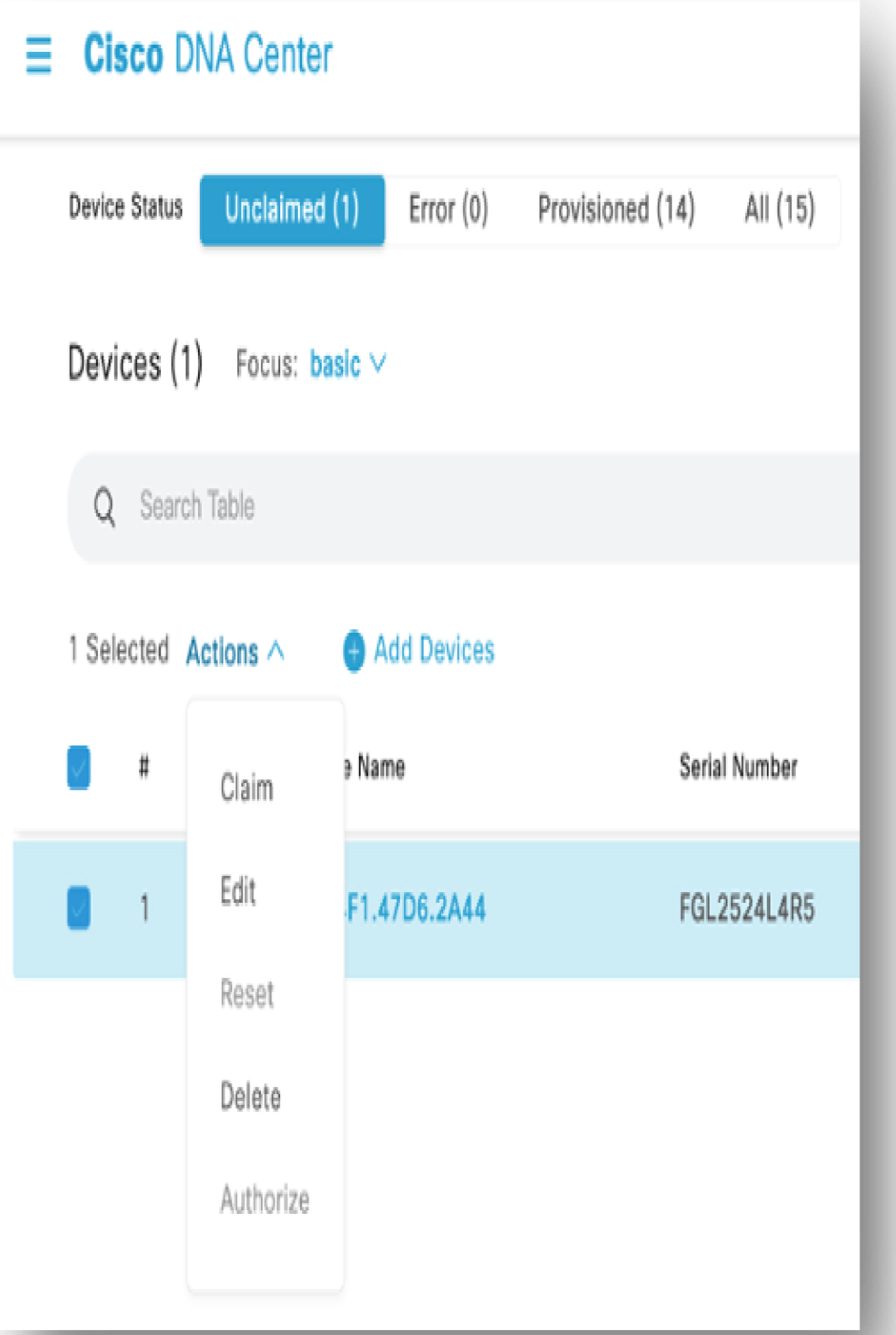


Figure 7-17 *Stack Cabling Schemas in Catalyst Center*

Claiming Devices in Catalyst Center

After the Plug-and-Play agent on a device has launched, and discovery has concluded, resulting in the device being in communication with its Plug-and-Play server (in this case, Catalyst Center), the device is in a state where it can be authorized and claimed by the operator. This operation can take place within the **Plug-and-Play** menu, as depicted in Figure 7-18.



☑	#	e Name	Serial Number
☑	1	F1.47D6.2A44	FGL2524L4R5

Figure 7-18 *Claiming a PnP Device*

Alternatively, you can proactively claim a device, supplying mandatory fields for hostname, serial number, and hardware model (PID). You can choose to perform this action on a single-device basis, through bulk import using CSV, or through a Smart Account. This action is also possible via the **Plug-and-Play** menu, as displayed in Figure 7-19.

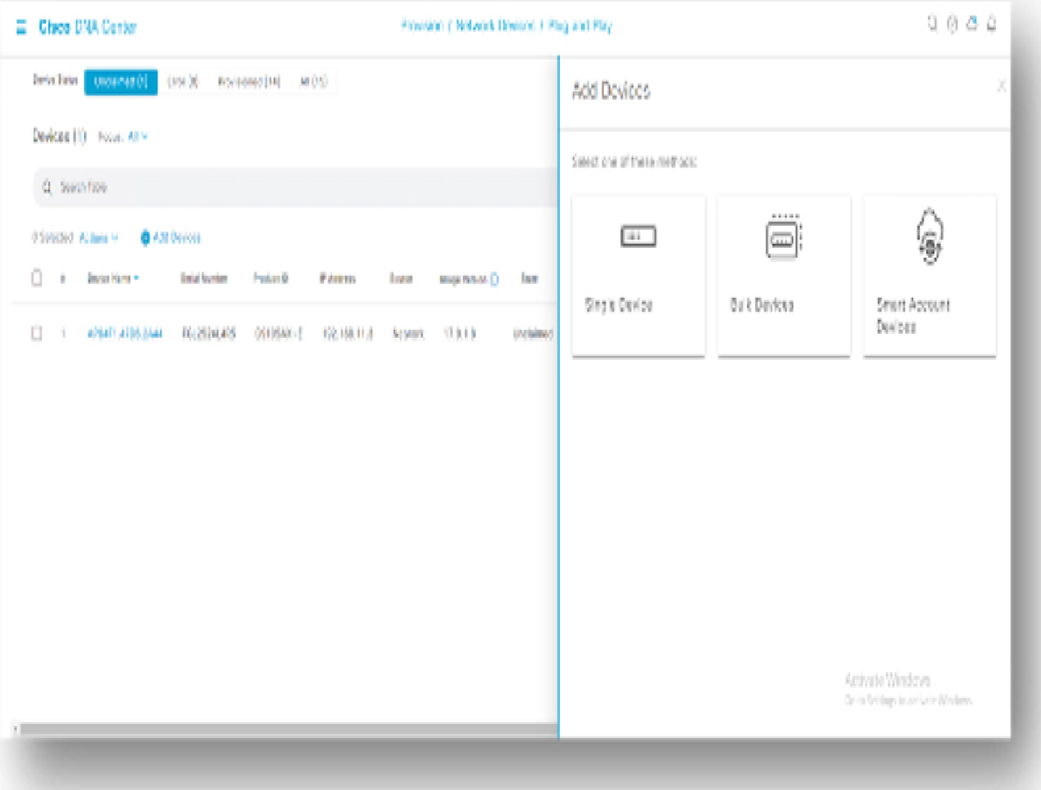
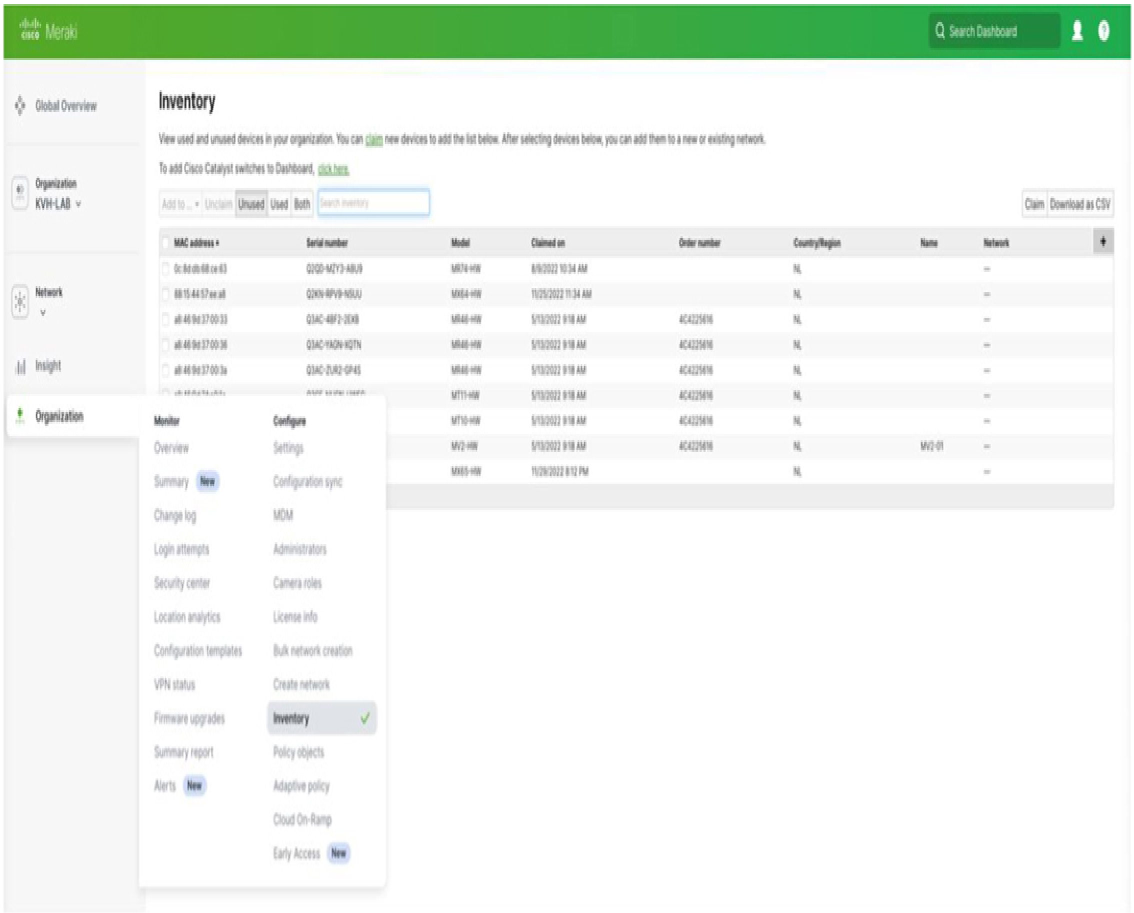


Figure 7-19 *Claiming Options*

Claiming Devices in the Meraki Dashboard

To claim devices in the context of Meraki components, you need to access the **Inventory** page on the dashboard, under the **Organization > Configure > Inventory**, as shown in Figure 7-20.



Inventory

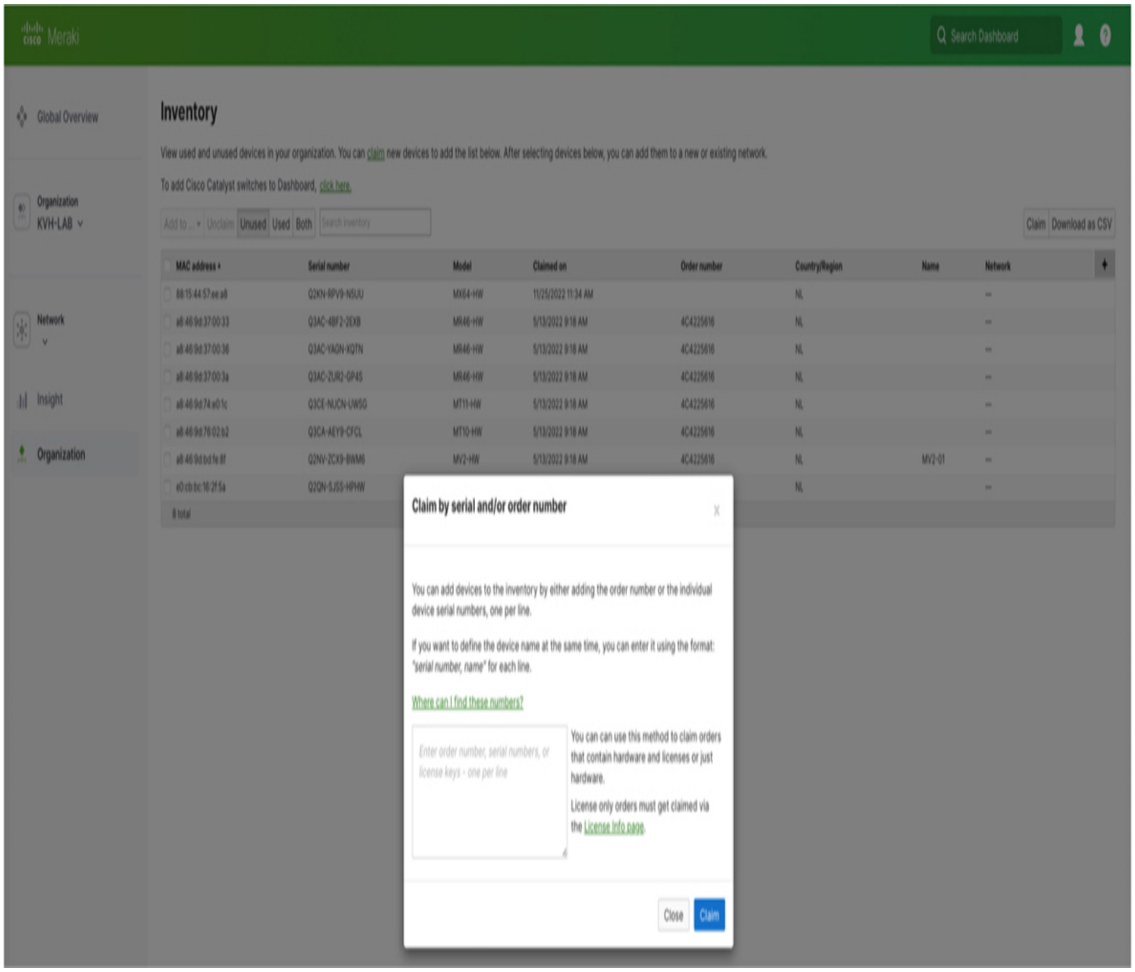
View used and unused devices in your organization. You can claim new devices to add the list below. After selecting devices below, you can add them to a new or existing network.

To add Cisco Catalyst switches to Dashboard, click here.

MAC address	Serial number	Model	Claimed on	Order number	Country/Region	Name	Network
0c:8d:db:68:ce:63	Q2QD-MZY3-A8U9	MR74-HW	8/9/2022 10:34 AM		NL		–
88:15:44:57:ee:a8	Q2KN-RPV8-NSUU	MX64-HW	11/25/2022 11:34 AM		NL		–
a8:46:9d:37:00:33	Q3AC-48F2-2EXB	MR46-HW	5/13/2022 9:18 AM	4C422516	NL		–
a8:46:9d:37:00:36	Q3AC-YAGN-XQTN	MR46-HW	5/13/2022 9:18 AM	4C422516	NL		–
a8:46:9d:37:00:3a	Q3AC-ZUR2-GP4S	MR46-HW	5/13/2022 9:18 AM	4C422516	NL		–
[illegible]	[illegible]	MT11-HW	5/13/2022 9:18 AM	4C422516	NL		–
		MT10-HW	5/13/2022 9:18 AM	4C422516	NL		–
		MV2-HW	5/13/2022 9:18 AM	4C422516	NL	MV2-01	–
		MX65-HW	11/28/2022 8:12 PM		NL		–

Figure 7-20 *Meraki Inventory Configuration Page*

You can claim a device using the serial number or the respective order number. Once claimed, as shown in Figure 7-21, the devices can be used within the organization.



Inventory

View used and unused devices in your organization. You can claim new devices to add the list below. After selecting devices below, you can add them to a new or existing network.

To add Cisco Catalyst switches to Dashboard, click here.

MAC address	Serial number	Model	Claimed on	Order number	Country/Region	Name	Network
88:15:44:57:ee:a8	Q2KN-RPV9-N5UU	MX64-HW	11/25/2022 11:34 AM		NL		–
a8:46:9d:37:00:33	Q3AC-4BF2-2EXB	MR46-HW	5/13/2022 9:18 AM	4C4225616	NL		–
a8:46:9d:37:00:36	Q3AC-YAGN-XQTN	MR46-HW	5/13/2022 9:18 AM	4C4225616	NL		–
a8:46:9d:37:00:3a	Q3AC-ZUR2-GP4S	MR46-HW	5/13/2022 9:18 AM	4C4225616	NL		–
a8:46:9d:74:e0:1c	Q3CE-NUCN-UWSG	MT11-HW	5/13/2022 9:18 AM	4C4225616	NL		–
a8:46:9d:76:02:b2	Q3CA-AEY9-CFCL	MT10-HW	5/13/2022 9:18 AM	4C4225616	NL		–
a8:46:9d:bd:fe:8f	Q2NV-ZCX9-BWM6	MV2-HW	5/13/2022 9:18 AM	4C4225616	NL	MV2-01	–
e0:cb:bc:16:2f:5a	Q2QN-SJS5-HPHW				NL		–

8 total

Claim by serial and/or order number

You can add devices to the inventory by either adding the order number or the individual device serial numbers, one per line.

If you want to define the device name at the same time, you can enter it using the format: "*serial number, name*" for each line.

Where can I find these numbers?

Enter order number, serial numbers, or license keys - one per line

You can can use this method to claim orders that contain hardware and licenses or just hardware.

License only orders must get claimed via the License info page.

Figure 7-21 *Claiming Meraki Devices*

Template Usage in Catalyst Center

The use of templates in Catalyst Center provides a means to deploy configurations that match the standard requirements of the organization. These configurations can range from day zero provisioning characteristics that are used simply to bring a device under network monitoring until further steps are applied, or they can be complete and final configurations for the nodes to operate in a completely provisioned and "gold standard" state, matching with the corporate requirements for both standardizations, based on site types, often referred to as *T-shirt sizes* or *security rulesets*, which are advised by the corporate security organizations.

Template languages that can be used on the Catalyst Center platform are Jinja2, which is a common language used by web programmers, or Velocity, which also has roots in web programming with Apache and was adopted in Cisco Prime Infrastructure (which, at the time of writing, is end of sale).

> **Note**
>
> While the learning curve for it is perhaps a little bit steeper, Jinja2 should be considered the preferred option because it has a lot more flexibility and capability than Velocity.

When you are evaluating template usage, the first important decision that you will need to make is *standard* or *composite*. There is no real right or wrong answer when it comes to choosing the templating option; the choice really comes down to which options are the easiest for the functions at hand, with a perspective of now and in the future.

Standard Templates

For most beginners, standard templates are probably the most simplistic way to get started. A standard template is comparable to a single script or recipe where all the information is required on a single page. While this may sound rather limited, through the addition of customizable variables and loops, more elaborate routines can be created for deployment.

The following example shows how you could configure a standard template to provision a basic function using the velocity scripting language:

```
!
hostname $HOSTNAME
!
```

The following is a comparable example written in Jinja2:

```
!
hostname {{ HOSTNAME }}
!
```

The format is straightforward: The IOS configuration parameter hostname is provided together with a variable represented by a unique name that is front-delimited with a $ symbol when using Velocity or delimited on both sides between curly brackets {{ }} in Jinja2.

When you’re applying such variables in a template, Catalyst Center will prompt you to populate the respective variable translation for the specific node that is being provisioned at the time of provisioning.

Composite Templates

While many network operators and site reliability teams are content using standard templates, when we’re looking at larger environments at scale, the lack of modularity can often result in duplicity, resulting in many large template files being created with minor differences to address each use case.

The use of composite templates takes an approach that is closer to object-oriented programming, whereby each function is separated into a separate template file that, through the modularity (illustrated in Figure 7-22), can be pieced together to represent the correct components to be deployed to a specific node.

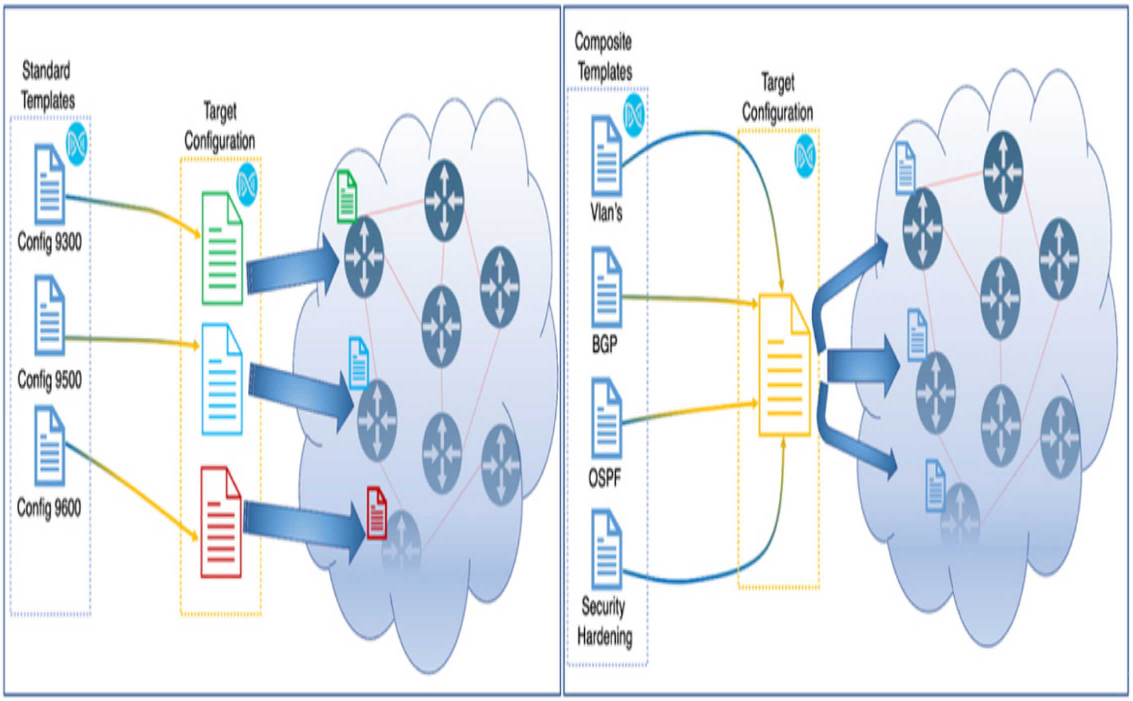


Figure 7-22 *Differences Between Standard and Composite Template Usage*

Consider a use case where composite templates create a more simplistic deployment capability; in this case, multiple platform types are in use. The use of the composite templates increases the reusability of templating code due to the nuances of variation between one platform and another.

> **Note**
>
> To properly deep-dive into templates within Cisco Catalyst Center, we recommend evaluating existing templating information in the GitHub repo at https://github.com/kebaldwi/DNAC-TEMPLATES. This information is from one of our talented colleagues, Keith Baldwin. In this repository, you can find labs, tutorials, and sample code for templates, EEM scripts, and REST API collections that cater to wired and wireless use cases.

Bouncing Interfaces

During the Plug-and-Play process, on switches, VLAN 1 will be used as the default entry VLAN for the Plug-and-Play process to initiate. While this configuration is very helpful for the initial onboarding of the device, it may not be the final and target configuration that is desired for the deployment. Often, during the provisioning phase, a change of VLAN identifier is desired to allow an onboarded device to transition between a temporary VLAN and IP address allocation toward a static IP address allocation that links up with the rest of the network infrastructure. Upon conclusion of Plug-and-Play, this change also results in the device initiating the HTTPS communication channel using a different update-source interface.

This action through the use of a template configuration—similar to what is shown in Example 7-4—ensures that the new interface that is configured for the management of the system (often a loopback) will conclude the PnP onboarding process and become the resultant management IP address within the PnP server.

Example 7-4 *Interface Bounce Template*

```
!
interface Vlan1
```

```
shutdown
!
<Further Template Content>
!
interface Vlan1
no ip address dhcp
no shutdown
!
```

Programmability-Based Deployment

Within service provider and enterprise environments, it is becoming more and more common to see customers decide to use DevOps principles from the software development world and apply them to the network. The use of NetDevOps and Infrastructure as Code (IaC) provides network operators with options that can represent a reduction in expenditure, stability, and time to deliver in a world where physical networks are often perceived as the block in expedient service deployment.

An example of how a programmability approach can be used to achieve the deployment of composite Plug-and-Play templates is shown in Figure 7-23.

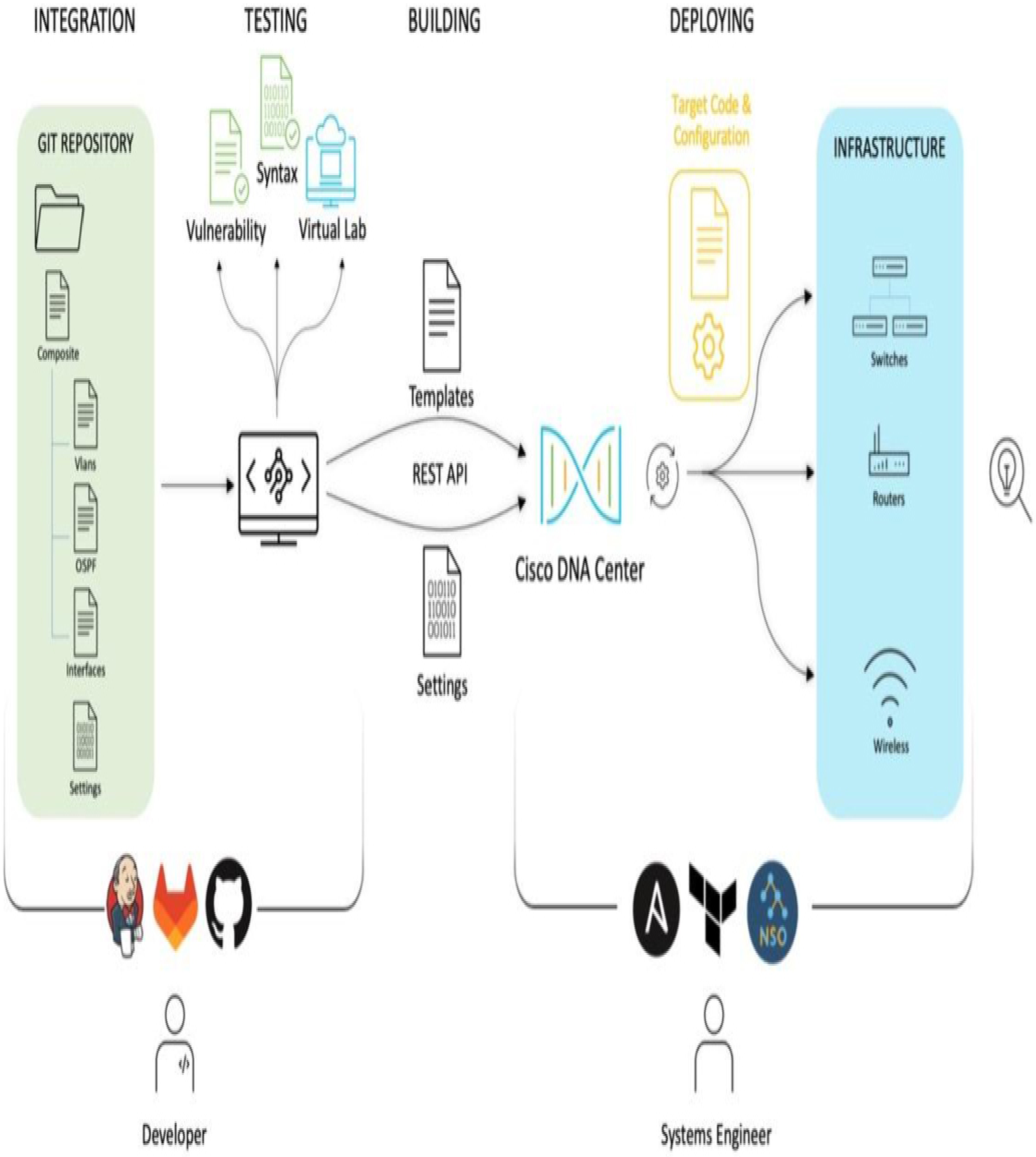


Figure 7-23 *An Infrastructure as Code Approach to Composite Templates with Catalyst Center*

Although this book does not focus primarily on this area, some of the programmability-based tasks that are relevant to speeding up and building a secure and resilient deployment through programmability are covered in the following section.

For a deeper dive into these topics, we highly recommended *Model-Driven DevOps* by Steven Carter and Jason King (Addison-Wesley Professional), *Network Automation Made Easy* by Ivo Pinto (Cisco Press), and

Automating and Orchestrating Networks with NetDevOps by Ivo Pinto and Faisal Chaudhry (Cisco Press).

Using Direct API Calls to Claim Devices

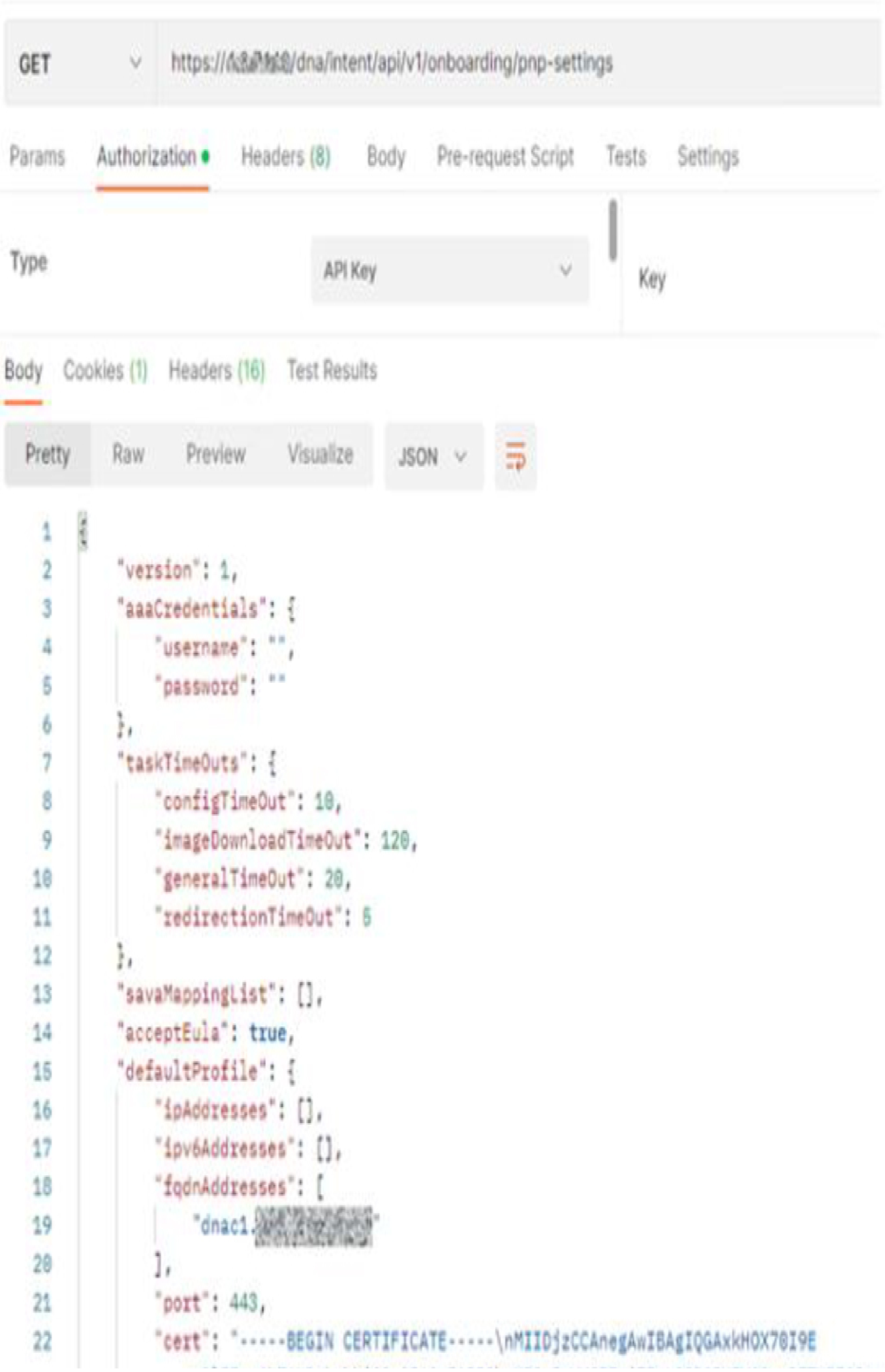
Many customers are beginning to shift their operating procedures toward an Infrastructure as Code approach, which has been very common in the cloud/hyperscaler world. Although using APIs instead of the GUI can, at times, be perceived as the most complicated way to achieve things, the use of direct API calls opens the door to automation at scale and an ability to build out functions that could be linked into more complex activities in the context of business process automation. For example, these activities include linking in with billing systems, or updating and informing customers of an activation via a chatbot on applications like Webex Teams, Slack, or Microsoft Teams.

In the context of onboarding, Figure 7-24 to Figure 7-28 show the execution steps performed using the Postman application.

The following steps outline how to use APIs for Plug-and-Play onboarding activities for an AP.

Step 1. GET

```
/dna/intent/api/v1/onboarding/pnp-settings
```



```
{
    "version": 1,
    "aaaCredentials": {
        "username": "",
        "password": ""
    },
    "taskTimeOuts": {
        "configTimeOut": 10,
        "imageDownloadTimeOut": 120,
        "generalTimeOut": 20,
        "redirectionTimeOut": 5
    },
    "savaMappingList": [],
    "acceptEula": true,
    "defaultProfile": {
        "ipAddresses": [],
        "ipv6Addresses": [],
        "fqdnAddresses": [
            "dnac1.[redacted]"
        ],
        "port": 443,
        "cert": "-----BEGIN CERTIFICATE-----\nMIIDjzCCAnegAwIBAgIQGAxkHOX70I9E
```

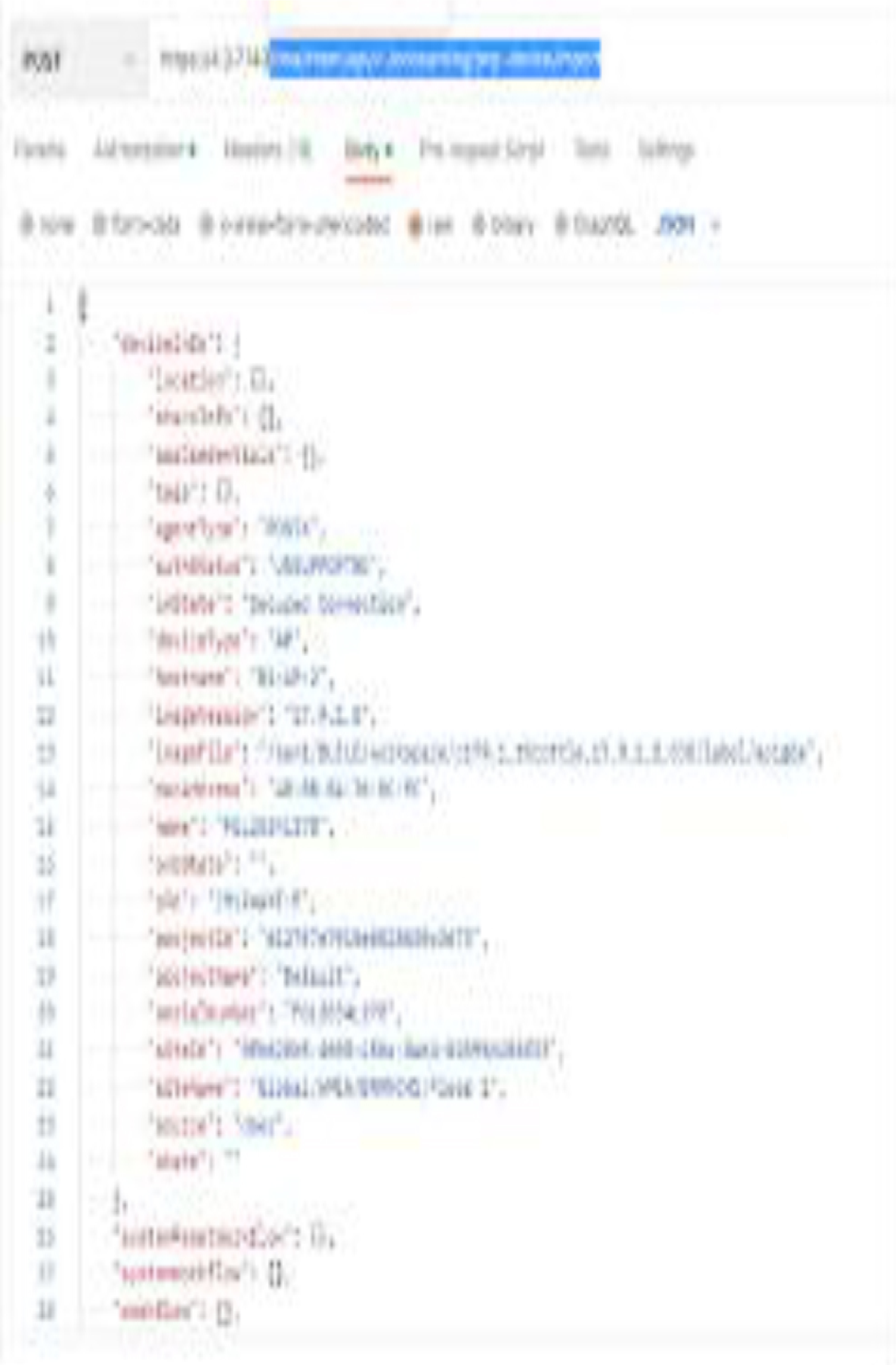
```
+9lFFym6kTANBgkqhkiG9w0BAQsFADBO\nMRQwEgYKCZImiZPyLGQBGRYEY29ycDETMBEGCg
JkiaJ\nk/IsZAEZFgNzYXAxITAfBgNVBAMTGHNhcC1TSEFSRURTRVJWSUNFU1ctQ0EtMTCC\n
```

Figure 7-24 *GET PnP Onboarding Overview*

Figure 7-24 provides an overview of Plug-and-Play settings, detailing information such as the EAP type for the AP onboarding flows.

Step 2. POST

```
/dna/intent/api/v1/onboarding/pnp-device/import
```



https://[illegible]

```
{
    "deviceInfo": {
        "location": [],
        "sharedInfo": [],
        "[illegible]": [],
        "tags": [],
        "agentType": "POSIX",
        "authStatus": "UNSUPPORTED",
        "inState": "[illegible]",
        "deviceType": "AP",
        "hostname": "[illegible]",
        "imageVersion": "[illegible]",
        "imageFile": "[illegible]",
        "macAddress": "[illegible]",
        "name": "[illegible]",
        "[illegible]": "",
        "pid": "[illegible]",
        "projectId": "[illegible]",
        "projectName": "Default",
        "serialNumber": "[illegible]",
        "siteId": "[illegible]",
        "siteName": "[illegible]",
        "source": "User",
        "state": ""
    },
    "[illegible]": [],
    "systemWorkflow": [],
    "workflow": [],
```

Figure 7-25 *Import Device into PnP*

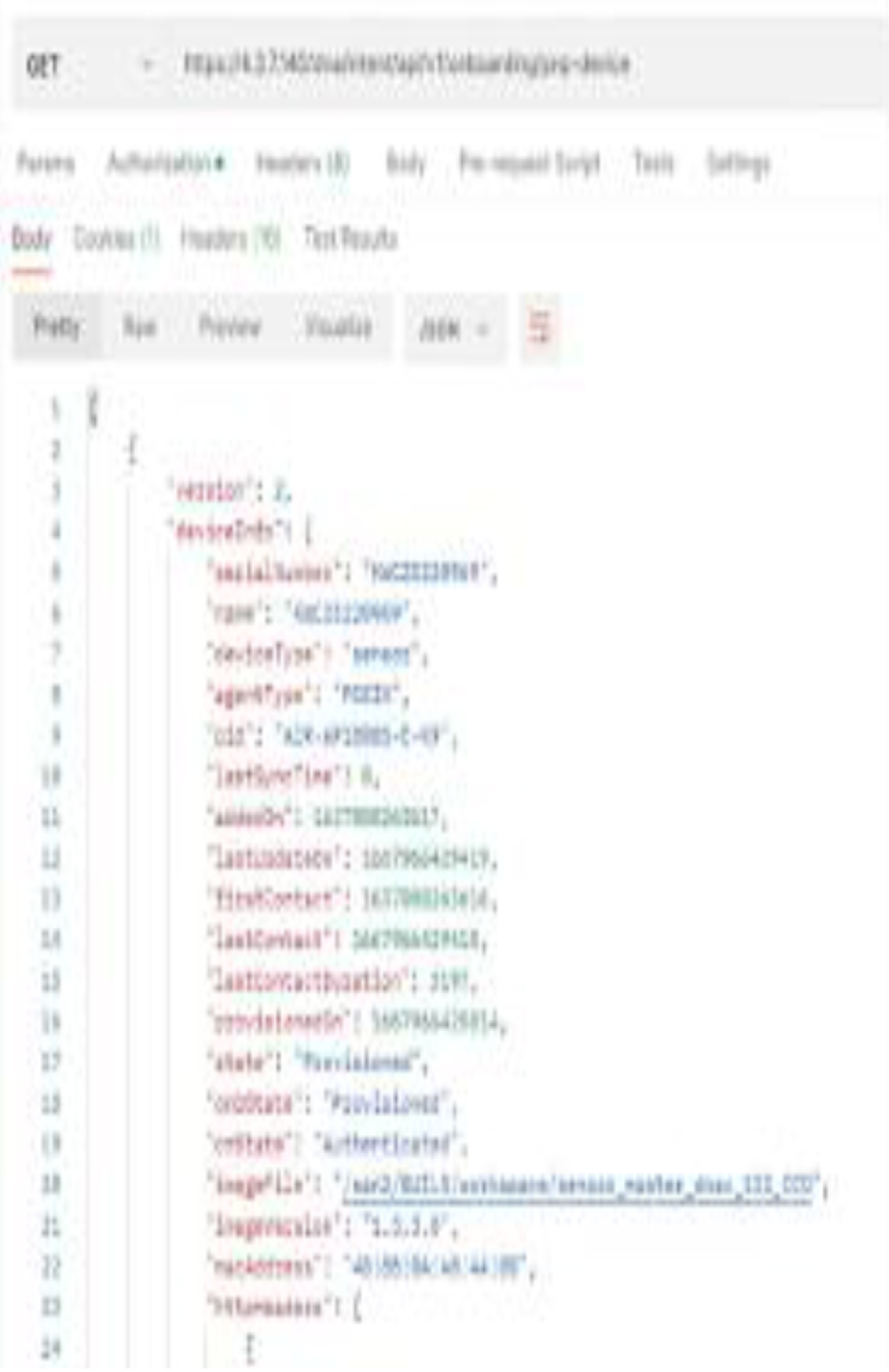
The API post in Figure 7-25 is used to import a device to be claimed, including mandatory (red) and/or optional (green) information such as

- Hostname
- Serial Number
- Hardware Model (PID)
- Site Name
- AP Name
- Custom Image

The response ID (not depicted) is used from the API call in Figure 7-27 for ssubsequent actions.

Step 3. GET

```
/dna/intent/api/v1/onboarding/pnp-device
```



```
[
    {
        "version": 2,
        "deviceInfo": {
            "serialNumber": "[illegible]",
            "name": "[illegible]",
            "deviceType": "sensor",
            "agentType": "POSIX",
            "pid": "AIR-AP1800S-E-K9",
            "lastSyncTime": 0,
            "addedOn": [illegible],
            "lastUpdateOn": [illegible],
            "firstContact": [illegible],
            "lastContact": [illegible],
            "lastContactDuration": [illegible],
            "provisionedOn": [illegible],
            "state": "Provisioned",
            "onbState": "Provisioned",
            "cmState": "Authenticated",
            "imageFile": "[illegible]",
            "imageVersion": "[illegible]",
            "macAddress": "[illegible]",
            "httpHeaders": [
                {
```

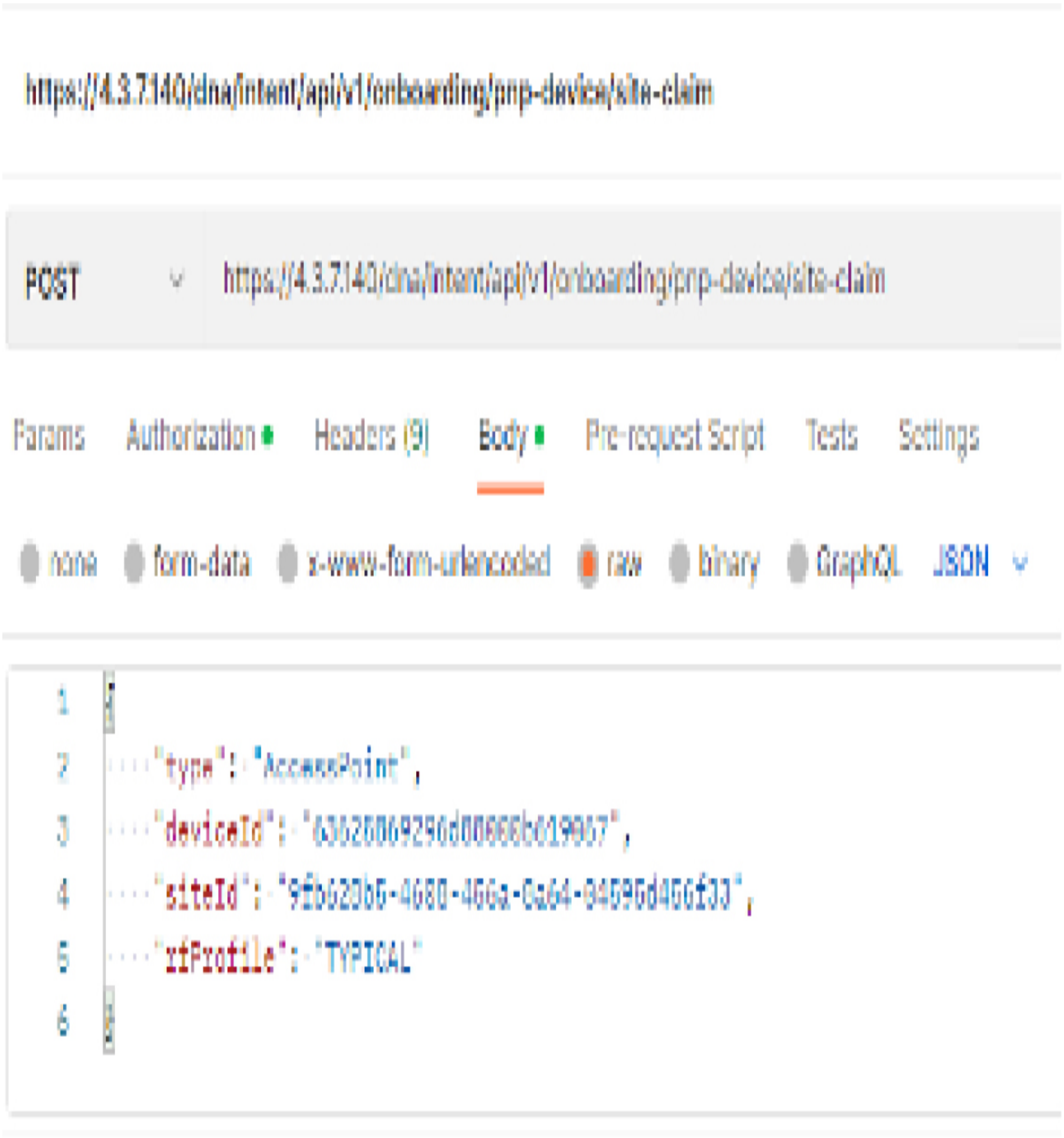
```
                    "key": "clientAddress",
                    "value": "10.27.100.196"
                }
            ],
```

Figure 7-26 *Verification of Imported Devices into PnP*

Verifying imported PnP devices by using a GET call in Figure 7-26, you can get a view of all imported devices. This includes all possible Plug-and-Play devices.

Step 4. POST

```
/dna/intent/api/v1/onboarding/pnp-device/site-claim (claim
Reference source not found.7)
```



https://4.3.7.140/dna/intent/api/v1/onboarding/pnp-device/site-claim

POST https://4.3.7.140/dna/intent/api/v1/onboarding/pnp-device/site-claim

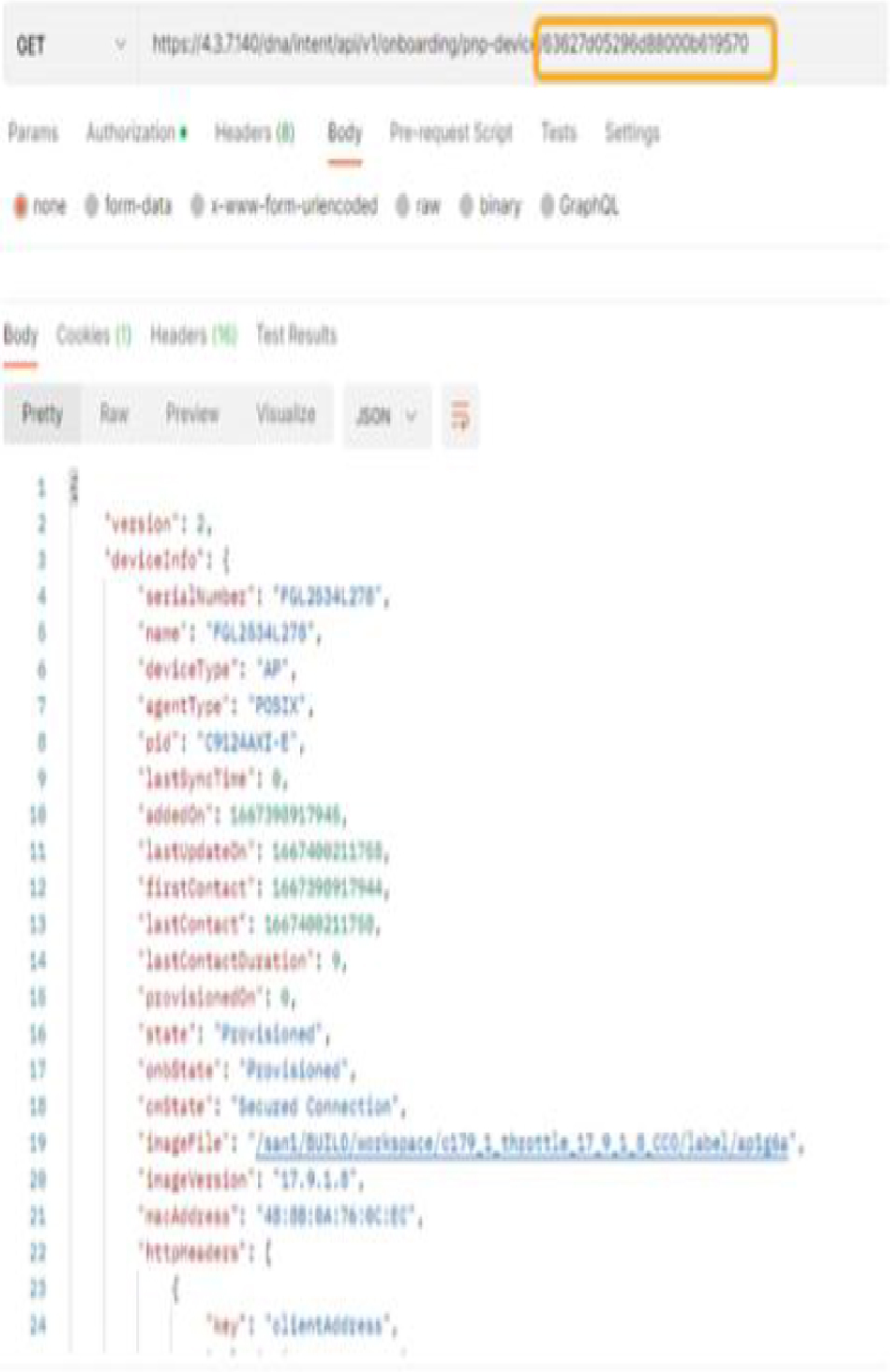
```
{
    "type": "AccessPoint",
    "deviceId": "63626069296d00006b019067",
    "siteId": "9fb620b6-4680-466a-0a64-04596d466f33",
    "rfProfile": "TYPICAL"
}
```

Figure 7-27 *Claiming Imported Devices via API*

To claim imported devices, you can use the Site ID and deviceID in Figure 7-27 to claim an imported device.

Step 5. GET

```
/dna/intent/api/v1/onboarding/pnp-device
```



https://4.3.7.140/dna/intent/api/v1/onboarding/pnp-device/63627d05296d88000b619570

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

```
{
    "version": 2,
    "deviceInfo": {
        "serialNumber": "FGL2534L278",
        "name": "FGL2534L278",
        "deviceType": "AP",
        "agentType": "POSIX",
        "pid": "C9124AXI-E",
        "lastSyncTime": 0,
        "addedOn": 1667390917946,
        "lastUpdateOn": 1667400211758,
        "firstContact": 1667390917944,
        "lastContact": 1667400211750,
        "lastContactDuration": 0,
        "provisionedOn": 0,
        "state": "Provisioned",
        "onbState": "Provisioned",
        "cmState": "Secured Connection",
        "imageFile": "/san1/BUILD/workspace/c179_1_throttle_17_9_1_0_CCO/label/ap1g6a",
        "imageVersion": "17.9.1.0",
        "macAddress": "48:0B:0A:76:0C:EC",
        "httpHeaders": [
            {
                "key": "clientAddress",
```

Figure 7-28 *Post Claim Verification*

You can verify the status of a claimed device via this API call, as shown in Figure 7-28.

The state that is viewed within the output transitions from Unclaimed to Provisioned.

API states associated with PnP are

- Unclaimed
- Planned
- Onboarding
- Provisioned
- Errored

Claiming Devices Using Ansible

Ansible has become a key tool in the toolbox for network architects who are starting the journey toward automation within their environments. Through its use, within a very short period of time, you can derive value by providing an abstraction between code and scripts, without the need to be a programmer.

In the following sections, we will explore how to use Ansible as a means to perform the claiming operation. This is done using the existing Meraki and Catalyst Center SDKs that exist within Ansible Galaxy.

For the deployment of Catalyst Center[nd]based Ansible playbooks, you can use the following collection. At the time of writing, the repository was well maintained:

https://galaxy.ansible.com/cisco/dnac

Issues identified with the repo should be raised via GIT—not through the Cisco Technical Assistance Center.

Customer Use Cases

Throughout this chapter, we have described Plug-and-Play as a means to allow organizations to reduce the time it takes to deploy, roll out, and scale their environments in a structured manner. When coupled with automation, this approach can lower overhead and potentially reduce the margin of error for deployments.

The following sections describe two customer use cases, where Plug-and-Play and its corresponding tools have helped organizations reduce the time to value through its deployment.

Large Banking Customer: Pan-Africa Deployment

If you are familiar with the Catalyst 9000 Series switches, you may have observed a bright blue light on the top left corner of the device (see Figure 7-29). This blue LED was deployed in the C9K with the intention to allow lab and implementation teams who are in a remote server rooms to be able to find a switch in a rack while the operator may be remotely making changes. This capability has saved countless network deployments from having cables pulled out on the incorrect switch.

To activate this light, you can execute the **hw-module beacon slot 1** command from the enable mode CLI of a switch. The off syntax will deactivate this feature.

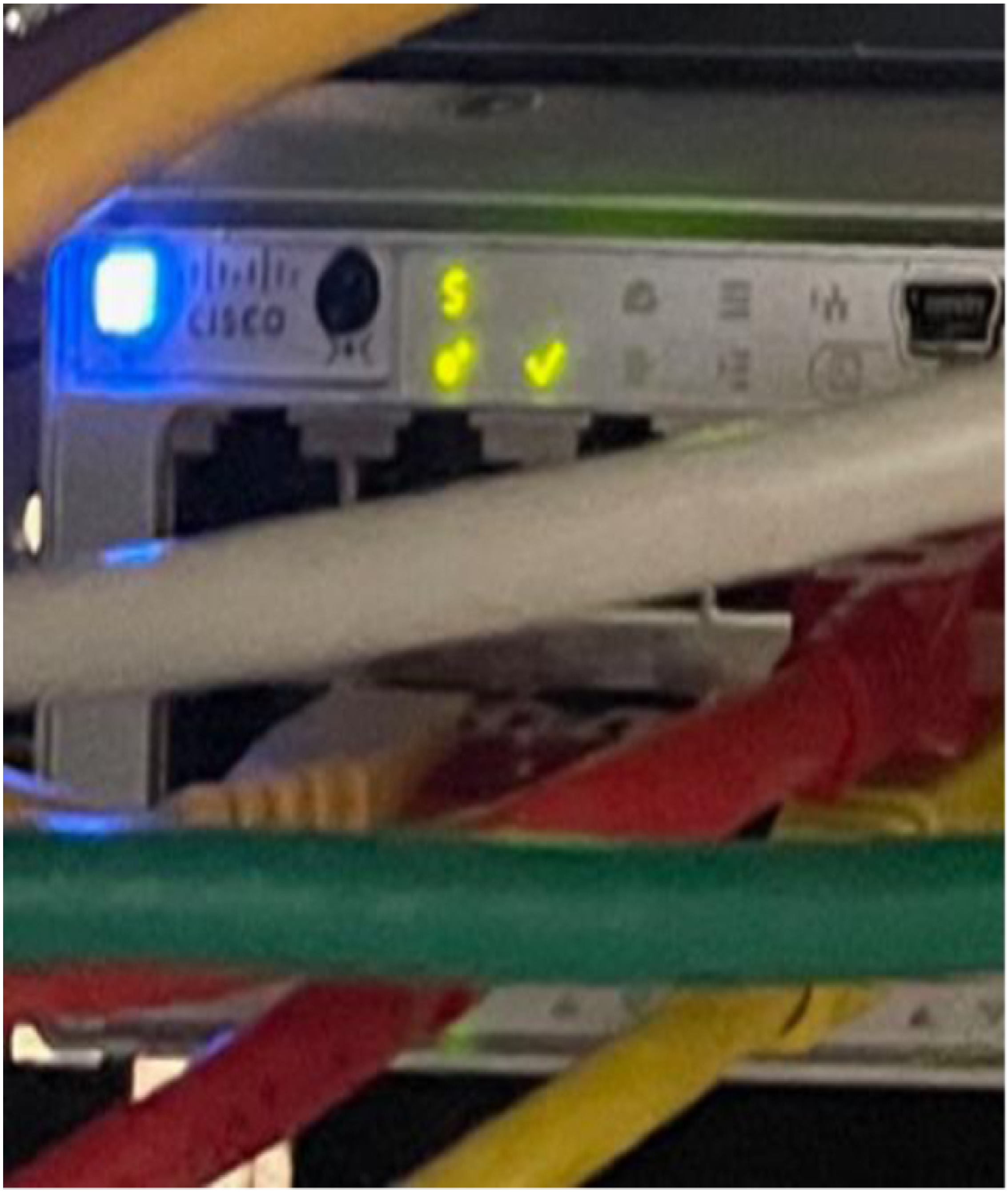


Figure 7-29 *Blue Configurable Beacon on a Catalyst 9300 Series Switch*

By now, you are probably thinking, Why we are talking about the blue LED for locating a switch in a chapter that is focused on Plug-and-Play? The reason here is quite simple: Although this LED can, and should, be used for the intention that was described earlier, it doesn't have to be used only for this purpose.

In 2024, Cisco Professional Services (CX) team members were engaged in a project with a large banking customer based in Africa. The customer had branch locations all across the continent and was at the beginning of evaluating how it could begin a networkwide refresh. One of the customer's challenges was the lack of skilled personnel in some locations that were able to configure a router or a switch. In some places where the customer had branches, the locations were not particularly safe, which made the existing skilled workforce reluctant to travel to support upgrades and migration activities. IT activities in many of these sites were often handled by an IT manager, who was a jack-of-all-trades, responsible for many activities beyond just IT within these locations. To address this challenge, the customer's executives turned to Cisco for support in finding a solution.

The Cisco team took a look at the topology requirements for these branch sites; they were all connected via a Cisco SD-WAN solution. For the end-to-end network to function, the SD-WAN C-Edge devices needed to connect the wide area network and establish BGP peerings with the new branch routers to bring the network into service. To achieve these actions, several steps needed to take place. A failure in one of the steps would result in the branch not being configured.

To deploy the branch, the team came up with a simple flow diagram, which detailed how the state of the location should look and which steps would need to be involved in the respective execution.

Step 1. Provision a temporarily routable IP range on the C-Edge in VPN 43.

Step 2. Configure an IP Helper address on the Layer 3 interface of the routable range.

Step 3. Configure a DHCP server pool for temporary addressing, including Option 43 to point to the PnP server (in this case, Catalyst Center)

Step 4. Preconfigure a day zero template on Catalyst Center, including a conditional ruleset for LED activation upon BGP peering being established.

Step 5. Preclaim a serial number and device to location, allocating the correct location.

Following these steps, the key network systems were primed for the arrival of new switches to be installed in their remote sites.

Example 7-5 shows a copy of a similar day zero PnP script that was used for the onboarding (written in Velocity).

Example 7-5 *Plug-and-Play Script for LED-Based Signaling*

```
!
event manager applet catchall
event cli pattern ".*" sync no skip no
action 1 syslog msg "#[[$_cli_msg]]#"
!
event manager applet
event syslog pattern " %BGP-5-ADJCHANGE: neighbor.* Up"
action 1 syslog msg "C-Edge Connection ESTABLISHED - VPN 43 Manua
action 1 cli command "hw-module beacon slot 1 on"
!
ip routing
!
hostname $HOSTNAME
!
interface Vlan1
shutdown
!
vlan 43
name INFRA_VN_VIA_SDWAN
!
interface Vlan43
ip address $LOGICAL_VLAN43_FIAB 255.255.255.252
no shut
!
```

```
interface Loopback0
 ip address $LOOPBACK0 255.255.255.255
!
interface GigabitEthernet1/1/1
description UPLINK_TO_CEDGE
switchport mode trunk
switchport trunk allowed vlan 2-4094
!
ip http client source-interface loopback 0
!
ip ssh source-interface loopback 0
!
router bgp 65001
bgp router-id $LOOPBACK0
bgp log-neighbor-changes
neighbor $BGP_UPSTREAM_NEIGHBOR remote-as 65002
neighbor $BGP_UPSTREAM_NEIGHBOR update-source vlan43
!
address-family ipv4
network $LOOPBACK0 mask 255.255.255.255
neighbor $BGP_UPSTREAM_NEIGHBOR activate
exit-address-family
!
interface Vlan1
no ip address dhcp
no shutdown
!
no event manager applet catchall
!
do hw-module beacon slot 1 off
!
end
```

Upon deployment of the first site, the customer operations team waited eagerly to see if the installation of one of the new devices took place without a hitch. To their surprise, the first branch had new equipment come online without intervention, with a call from the branch that the blue LED light had turned on, which could be confirmed by the operations team.

Global Deployment: Large-Scale Enterprise Deployment

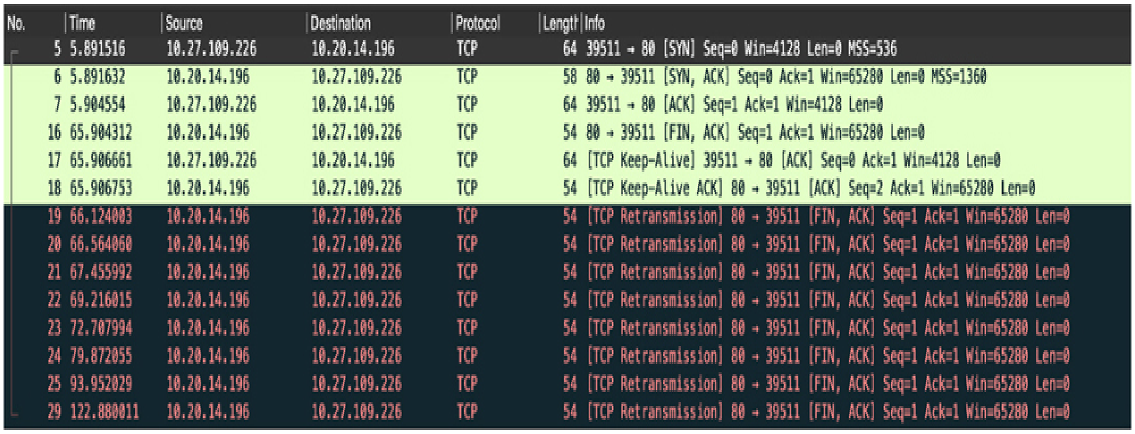
The Cisco CX team members also worked on another large global deployment. This time it was for a global enterprise software company that was in the process of deploying 400 sites globally using LISP-based SD-Access, connecting a wide area network globally using SDA-Transit, which allows for overlay tunneling using VXLAN between sites, maintaining SGT and virtual network information. This customer decided for this approach for global deployment as a simplistic means to ensure that it can maintain the requisite end-to-end microsegmentation and macrosegmentation given that it was connecting to a third-party SD-WAN solution that didn’t properly support maintaining TrustSec, which would have severely limited security posture globally.

With a broad global deployment, the customer used a mix of SD-Access Fabric sites that were deployed using LAN automation, and other sites that were connecting extended node devices, which utilize Plug-and-Play onboarding to provision into the fabric network.

The flow for such devices usually is relatively straightforward:

1. Boot up in factory default mode and launch the PnP agent.
2. Receive a DHCP or DNS address with viable PnP server information (as described earlier).
3. Contact the PnP server via HTTP.
4. Wait to be claimed and onboarded (or automatically onboarded for supplicant based extended nodes (SBENs).

In the scenario for this specific customer, the onboarding process would never complete. Instead, the customer observed a three-way TCP handshake between the Catalyst Center Plug-and-Play server, but the connection failed as soon as HTTPS TLS negotiation began, as shown in the packet capture in Figure 7-30.



No.	Time	Source	Destination	Protocol	Lengtł	Info
5	5.891516	10.27.109.226	10.20.14.196	TCP	64	39511 → 80 [SYN] Seq=0 Win=4128 Len=0 MSS=536
6	5.891632	10.20.14.196	10.27.109.226	TCP	58	80 → 39511 [SYN, ACK] Seq=0 Ack=1 Win=65280 Len=0 MSS=1360
7	5.904554	10.27.109.226	10.20.14.196	TCP	64	39511 → 80 [ACK] Seq=1 Ack=1 Win=4128 Len=0
16	65.904312	10.20.14.196	10.27.109.226	TCP	54	80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
17	65.906661	10.27.109.226	10.20.14.196	TCP	64	[TCP Keep-Alive] 39511 → 80 [ACK] Seq=0 Ack=1 Win=4128 Len=0
18	65.906753	10.20.14.196	10.27.109.226	TCP	54	[TCP Keep-Alive ACK] 80 → 39511 [ACK] Seq=2 Ack=1 Win=65280 Len=0
19	66.124003	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
20	66.564060	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
21	67.455992	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
22	69.216015	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
23	72.707994	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
24	79.872055	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
25	93.952029	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0
29	122.880011	10.20.14.196	10.27.109.226	TCP	54	[TCP Retransmission] 80 → 39511 [FIN, ACK] Seq=1 Ack=1 Win=65280 Len=0

Figure 7-30 *Packet Capture of Broken PnP Flow*

Initially, the team troubleshooting the problem thought that the problem could be an MTU issue and started checking the path MTU between the extended node switch and Catalyst Center. To the team's surprise, the MTU was well within range, and not the source of the issue that the team had initially hypothesized. Looking into other avenues, the team started to look into two other areas that could have been causing the problem: a potential firewall rule blocking the communications or a potential routing issue.

Using the CLI commands on the switches and routers in the network, the operations team followed the exact path of the communications hop by hop using the **show ip cef exact-route** command to understand the forwarding flow. This exercise was performed in both directions, from source to destination and from destination to source. What this exercise revealed to the team was unexpected. Whereas the routed path to the wide area network was normal, leaving the branch site where the new extended node was planned to be onboarded, the return path from Catalyst Center to the extended node was traversing the global fabric network, being transported in VXLAN, from the main hub site, rather than directly over the WAN using the global OSPF routing instance.

Looking deeper into this issue, the team identified that this routing scenario came into play as a result of some incorrect summarization activities performed within the WAN, which made the hub LAN site more attractive to reach the extended node branch location in Munich than actually traveling over the wide area network. Although asymmetric routing is not always desirable, because it makes troubleshooting and operational continuity more difficult, it shouldn't be the actual reason for the Plug-and-Play process to not be working properly.

Now that the team properly understood the path, more in-depth troubleshooting was needed at a packet level. To troubleshoot these sorts of problems, it is important to understand the state of the packets in flight, at each hop throughout their journey. While this is not always feasible over a wide area network that is managed by a managed service provider or service provider, it should be possible within your own network.

In this scenario, we are looking at the topology shown in Figure 7-31.

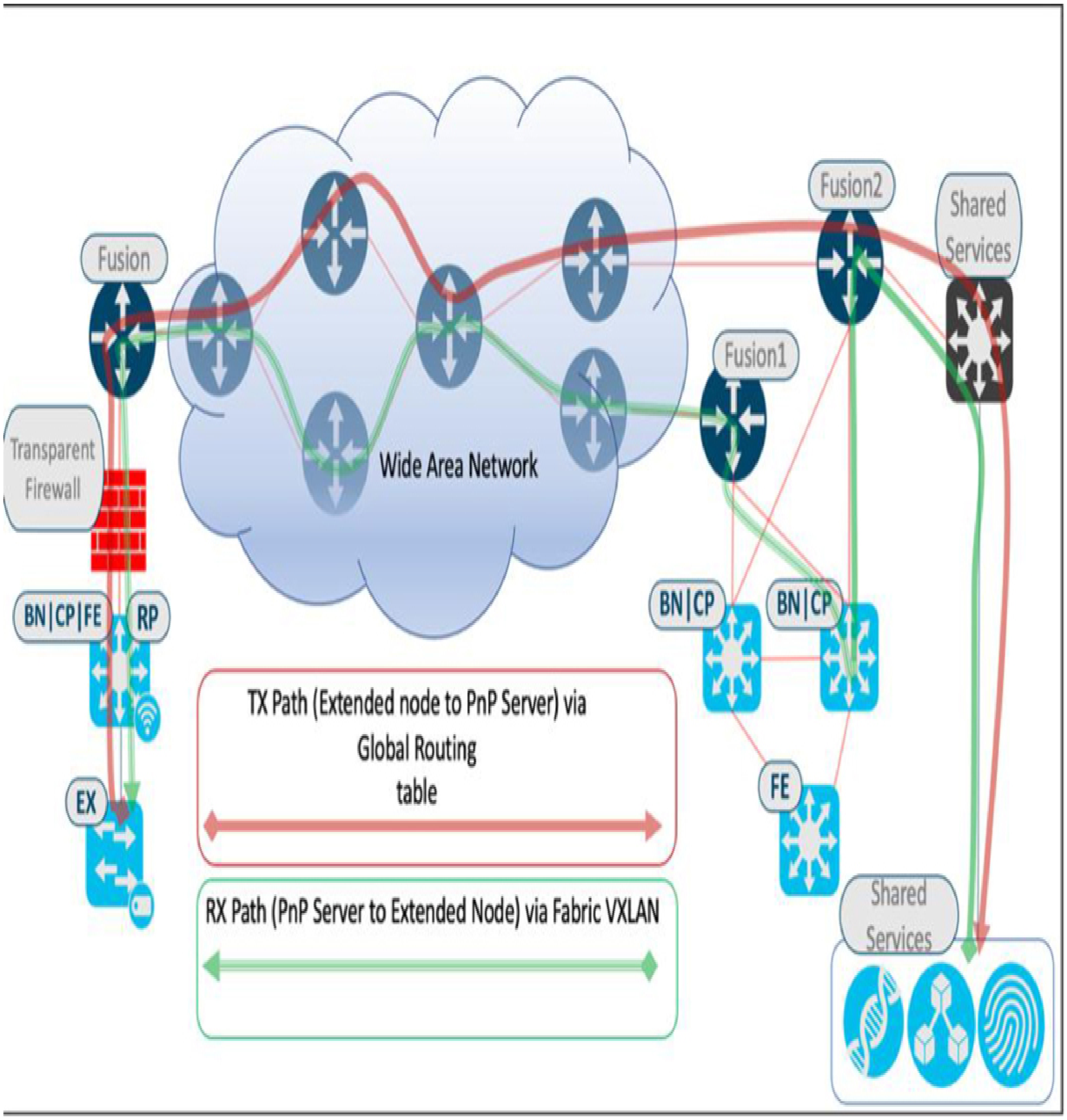

Fusion
Transparent Firewall
BN|CP|FE
RP
EX
Wide Area Network
Fusion2
Shared Services
Fusion1
BN|CP
BN|CP
FE
Shared Services
TX Path (Extended node to PnP Server) via Global Routing table
RX Path (PnP Server to Extended Node) via Fabric VXLAN

Figure 7-31 *Topology Displaying Asymmetric Routing*

When tracking each of these hops, we can see the payload of the traffic intact for most of the way. On the TX path from extended node to Catalyst Center, everything looks good. On the return path, it also looks fine, even over the wide area network on the CE router, indicating that the wide area network is not the culprit. What is interesting to see, however, is that between the CE and the fabric in a box (FIAB), certain packets are missing.

Initially, Cisco team members supporting the customer were confused. When they looked between those two devices, nothing appeared to be amiss. When they looked in CDP also, the neighboring device clearly could be seen and showed up as connected over Gig 1/0/1, but the packets were not making their way to the FIAB. After the Cisco team spoke with the customer's WAN team, they indicated that it might be a good idea to check with the security team again. Although they confirmed that their systems didn't have any rules blocking this sort of traffic, they had been known to have firewalls that operated in Layer 2 mode, and it would be good to confirm if one of these devices exited in the segment at the branch between the CE and FIAB.

After Cisco team members spoke to the customer's firewall team, they confirmed that they did indeed have a third-party firewall on that segment and wanted to take a closer look at the behavior now that they could confirm a difference in the packets between the two points. To their surprise, they identified that the packets were being dropped, but not by a standard rule, which they would expect; this explains why they didn't see any issues in the past. The packets were being dropped as a result of the asymmetric packet path; rules on the firewall resulted in the packets being silently dropped. Upon removal of that configuration, the team attempted the PnP process again, and this time, it worked successfully.

While this is not an everyday problem, it does raise a few important points in terms of steps needed for troubleshooting.

- When you're looking into differences in packet captures, it is always best practice to capture as close to the source and destination in parallel as possible (with synchronized clocking)
- Once captures have taken place, move in step by step until you can identify the last hop where the packet flow was intact.

Summary

Plug-and-Play represents a key tool in onboarding network infrastructure today in modern networks. Through the use of certificate-based validation and claiming authorization, devices are considered "untrusted" until such a point that they are authorized and onboarded into the network. Modern processes with Plug-and-Play provide a reliable means to scale up and deploy networks in smarter and more efficient ways, which are being more heavily adopted by Cisco customers around the globe.

Chapter 8. Routing and Traffic Engineering

In this chapter, you will learn the following:

- How to select an underlay routing protocol and understand its benefits and drawbacks
- How simple traffic engineering could make network more efficient and predictable
- How to secure routed networks
- How to operationalize an organization by ensuring less complexity but more flexibility in the routing architecture

Overview

The evolution of computer networks is one of the largest and fastest technological advancements in history. Even with the growth of power in computers and advancement of microprocessors, networks still have the upper hand. Today, all applications and hardware are built with one core element in mind: the need to communicate. This communication could be either within a contained system or information accessed from the cloud. There is always a need for an Internet Protocol (IP) packet to be sent out from a device at all times in today’s electronics.

Let’s look at the highlights from Cisco’s Annual Internet Report, as of 2023:

- There are 5.6 billion users actively using the Internet; that is nearly 66 percent of the global population.

- The number of devices connected to the Internet is 3.6 devices per capita; that is, for each person on the planet, there are 3.6 devices connected to the Internet.
- Many of the systems today are self-reliant and do not need human intervention; consequently, machine-to-machine (M2M) connections grew to 50 percent in 2023.
- The consumer segment will have nearly 74 percent of total device connections, with the business segment taking the remaining 26 percent. This number is staggering considering the use of smart homes, mobile devices, and many more daily consumer electronics.
- Over 70 percent of the global population was connected via mobile by the end of 2023.

The numbers in these highlights from Internet reports are staggering. Networks are evolving fast. The proliferation of the Internet of Things (IoT) has been a significant driver of this growth, with more and more devices such as smart home gadgets, wearables, connected cars, and industrial sensors coming online every day. With three times as many devices as the population of planet, the need to access information and communicate is crucial. Computers are individual components, but networks are connected systems, comprising massive clusters of computers and other electronics. Networks need to cater to all types of devices.

A better part of the success of today's highly resilient networks is due to the immense thought put into the design of routing traffic by employing a proper routing protocol strategy and traffic engineering. Routing protocols are crucial. Using the right one at the right place can make a network highly resilient and self-healing in the event of any outages. Similarly, traffic engineering goes hand in hand with routing protocols that provide optimization in the network and how all the connected and redundant links are utilized, increasing resiliency, efficiency, and better return on investment.

In a world where everything is connected, everyone—from enterprises, service providers, and residential users—plays a role. If you are connected to the Internet, you are the Internet. This global connectivity brings out good and bad people alike because they are all after the same goal: learning how the available information can be used for their own gain—regardless of whether it is good or bad. Take a pause and think about how this approach impacts the whole security and trust with what task someone is achieving on the Internet. For almost all humans, when they access the Internet on a daily basis, security is not top of their mind. Humans need to see things tangibly to see whether they are protected. When a human sees seat belts, warning signs, fences, or doors, for example, they relate those things to safety and security. In the cyber world, users don't see those things, and hence, they don't realize how much personal information is being made available to the general public. Just a simple "lock" icon on a website assures users that they are protected. But they do not have any idea how many years of work, standards development, and agreement between different organizations have gone into having that icon on a website so that users can feel safe.

How does this discussion relate to routing and traffic engineering? As the saying goes, we are as strong as our weakest link. If we look at the OSI layer, security cannot be applied on just one layer. It has to be applied across the stack. Just securing an application or putting a firewall in place does not make a network secure. There are a lot of moving parts, and all of those parts need to be considered. Layer 1 through Layer 7 are all important. Each layer has its own security model that can be scaled in various forms, and for the discussion in this chapter, let's concentrate on the Layer 3 aspect of it.

Routing is an essential part because it is what gets the network global. This is the first layer that extends beyond a local rack or broadcast domain, connecting different cities and the world. At the time of writing this chapter in October 2024, based on the IPv4 and IPv6 CIDR report, there were 984,244 (539,436 aggregated) IPv4 prefixes in the Internet routing table and 218,782 (10,980 aggregated) IPv6 prefixes in the Internet routing table. This is a massive number because each of these publicly routable prefixes will have thousands of endpoints. Securing access to them is another big challenge. Let's look now at what is involved in routing and traffic engineering security.

Routing

Routing has evolved over time. Just like how you need to know the route of your destination, today people rely heavily on the Global Positioning System (GPS) to route them to their destination in the most efficient way regardless of whether or not they have the route memorized. The old way of depending on an atlas is almost nonexistent. The dynamic information that comes in real time in terms of traffic conditions, road closures, construction, and so on is crucial for everyday commuters so that they can adapt and change course as necessary. These course corrections do not happen often, but knowing when to make them makes all the difference.

Similarly to human commute routing, the routing of packets in networks has also evolved. Computer networks have become vast and complex, with thousands of nodes in an enterprise network. Every bit of the detail in terms of network congestion, outages, and routing table changes is important in making meaningful decisions so that the end user experience is as seamless as possible. Today, if we look at a global network, hundreds of outages are happening every minute, and they also recover quickly as well. Mostly, all of this information is completely transparent to people due to the way routing and traffic engineering have been designed. Today, service providers are able to provide 99.999 percent (Five 9s) in their service-level agreements (SLAs) to customers at premium pricing, and the customers can also understand that in some businesses, network downtime is directly proportional to loss of business revenue. Predictability is paramount, and eliminating network outages and downtime is one of the most important variables in today's modern networks.

These highly efficient and resilient networks have multiple layers. Just as the saying goes—"Choose the right tool for the job"—using the correct routing mechanism in conjunction with redundancy and resiliency is an art that every network architect strives for. Just as with the evolution of people commuting, tools like static printed atlases, maps, and printed directions are obsolete, and over time, many of the legacy routing protocols have also been deprecated through evolution. Protocols such as the Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), and Policy-Based Routing (PBR) have been completely deprecated or replaced by newer protocols.

Now let’s look into the routing protocols and see how each of them has a role to play in a network and how they are suited for different network types.

From a software-defined networking point of view, most of the routing is divided into two main parts: underlay routing and overlay routing. Routing protocols are used to build the foundations of the network that can converge fast, whereas in the overlay part of the network, it is built for multitenancy, traffic segmentation, and scalability. It is like a fabric that can be stretched in layers on top of the network underlay. This potent combination of underlay and overlay networks, which can be all configured by some form of controller, is what modern networks are built of. This allows us to decouple transport and service. Figure 8-1 illustrates the concept of a modern SDN.

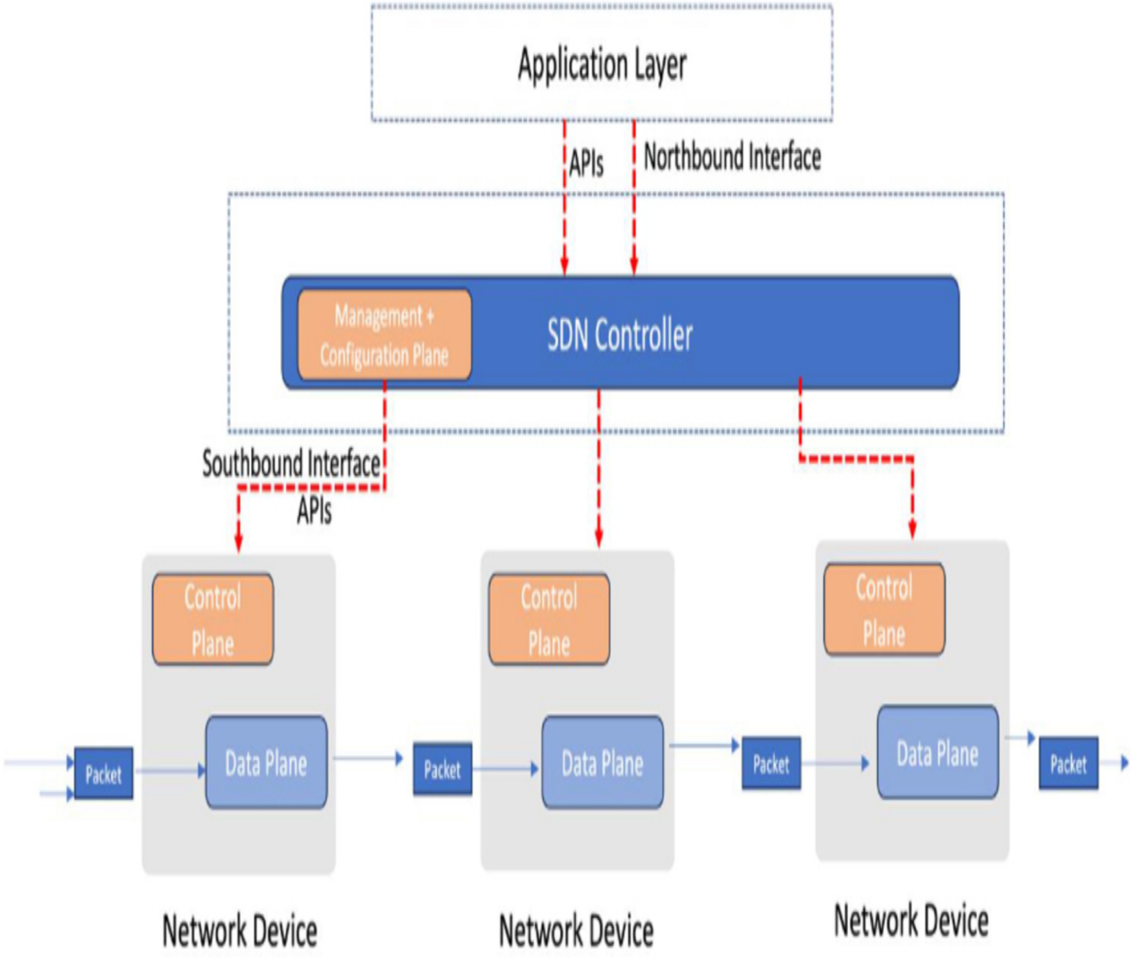


Figure 8-1 *Modern SDN*

Next, we will look at routing protocols differently compared to the traditional view. Having an overview of what protocols are used for the purpose and how we bridge the networks and overlays is crucial in understanding the ultimate goal of what they are serving.

Underlay Routing Protocols

A modern software-defined network has many layers to make it more resilient and secure. There is a data plane, control plane, policy plane, and management plane. All these planes work in conjunction with each other to provide the most robust of SDNs. This is not a new concept; we have been using this method for decades. With the evolution of technology and the ability to manage devices more effectively, we are seeing more software-driven architectures. Always be mindful of the fact that networks don't change overnight. It is not just a simple software upgrade on a server and you're up and running the newest features and abilities. On average, the refresh time for most networks is about 5[nd]6 years. It is not about getting maximum return on investment (ROI) on the network, but rather the complexity of the migration to a new technology. Networks are operating 24×7×365. Taking part of the network down is not easy, and because all devices are connected, you cannot just upgrade one device with new technology and leave everything else alone. A lot of planning takes place.

Here is one of the critical predicaments that every network owner faces: Take network downtime for an upgrade and add services or keep the network up and running OR don't be agile at deploying new services and leave the network as is. As the number of applications riding over the network has grown, there has been a huge requirement for networks to be agile. This need has changed the perspective on how we manage and maintain networks. One theory is that the SDN architecture was inspired by how large service providers manage their networks. Their networks were among the first ones that were highly scalable, agile, and serviceable. By using a multilayer approach on their networks, they were able to build a robust underlay architecture that decoupled from the overlay service architecture. This enabled them to continue with their routine device maintenance and also add new customers and services to the network on demand. Since most service providers have their own network management tools, they were—in some shape or form—using a software-driven network. The network being described here is Multiprotocol Label Switching (MPLS). By running multiple routing protocols, separation in the data plane, control plane, and policy plane was achieved. The management plane was achieved via automation tools.

Knowing the benefits of a separate data plane, control plane, and management plane in service provider networks, organizations began developing this type of fabric-based architecture for data center, campus, and WAN domains. Networks were not simple monoliths with single routing planes anymore. With the use of virtualization technology such as virtual routing and forwarding (VRF), on top of the underlying transport layer, the possibilities became endless. To this date, there are developments on how these virtual networks can be stretched to various sections of the organization's network.

All this change is possible because of robust underlay networks. For a network manager in an organization, it is difficult to balance network uptime and serviceability at the same time. The highest network uptime is achieved when the convergence of the network is fastest—subseconds. This can happen only when the network is routed. All Layer 3 links can converge fast and can incorporate advanced features like bidirectional forwarding detection (BFD). However, enabling routed networks up to the access layer is not efficient in terms of VLAN stretching. In a large organization or a massive building, there are multiple intermediate distribution frames (IDFs) across a given floor, and to ensure there is seamless Layer 2 connectivity for users, devices, and wireless roaming, it might not be possible to have Layer 3 up to the access layer. Here, the power of the overlay comes into play. Similar to a service provider's MPLS networks, the overlay networks can be offered on demand and usually do not have any impact on other overlay networks riding the same physical network device. This combination makes the network so versatile that, today, underlay and overlay networks are the norm for any network domain—whether data centers, campus, WAN, cloud, or security.

Underlay networks are usually built using any dynamic routing protocol. The ultimate objective of the underlay network is to provide high convergence capability with seamless failover. You cannot build a highly effective underlay network on nonredundant links. Hence, it is implied that redundancy is paramount for the underlying network to converge, because if the underlay fails, the overlay is sure to fail as well.

In today's resilient campus networks, the underlay can be built using any routing protocol—from static to BGP based. These protocols are not ideal, but depending on the scenario, they may work as well. But be mindful of why such protocols are needed. In most cases, static and BGP routing protocols are extremely rare. Static routes played a major role in small networks. Usually, they have been used to point to a default gateway and, in some cases, as a backup of dynamic routes in the form of floating static routes. In today's networks, besides a default gateway for a stub network, they are not used extensively. BGP is a powerful protocol; however, it was designed for the large-scale Internet and not for the campus network. It is slow to converge, and extreme fine-tuning is required for it to be fast converging. It defeats its own purpose at that level. Hence, modern IGPs are perfect for underlay networks. Service providers tend to rely on the Intermediate System to Intermediate System (IS-IS) routing protocol as their underlay because it provides a massive scale benefit. Enterprise networks, on the other hand, do not reach the scale limit that often, so it is not used. Also, there is a skill issue to address. Most enterprise network engineers and administrators are not familiar with the IS-IS routing protocol, so they have to rely on something fundamental and basic to achieve the underlay routing. Today's network controllers, such as Catalyst Center and Application Policy Infrastructure Controller (APIC), provide automated underlay provisioning, so it is an easy choice to use a scalable IS-IS routing protocol in the underlay.

Next, let's look at some of the most common routing protocols that build today's underlay networks.

Enhanced Interior Gateway Routing Protocol (EIGRP)

Enhanced Interior Gateway Routing Protocol is one of the fastest converging routing protocols out there. It is Cisco's proprietary routing protocol, which implies it works with Cisco's routers and switches. This protocol is an "advanced distance vector" protocol using Diffusing Update Algorithm (DUAL) that takes many attributes of a network into account before deciding on the best path. It is an enhancement of the Interior Gateway Routing Protocol (IGRP) and is designed to be more efficient and scalable. EIGRP uses a composite metric formula to calculate the best path to a destination. The composite metric is based on several factors (K values), including bandwidth (K_1), delay (K_2), load (K_3), reliability (K_4), and maximum transmission unit, or MTU (K_5). However, the default EIGRP metric calculation primarily uses bandwidth and delay:

$$Metric = \left[\left(K_1 \times BW + \frac{K_2 \times BW}{256 - Load} + K_3 \times Delay\right) \times \frac{K_5}{K_4 + Reliability}\right]$$

As you can observe, this protocol can be highly tuned to build efficient network routing.

Some of the key features of EIGRP are

- **Efficient Use of Resources:** EIGRP uses bandwidth and processor resources efficiently.
- **Rapid Convergence:** EIGRP quickly adapts to network topology changes, which minimizes downtime.
- **Loop-Free Operation:** EIGRP uses the diffusing update algorithm (DUAL) to ensure a loop-free routing environment at every instant.
- **Support for Multiple Network Layer Protocols:** EIGRP supports IPv4, IPv6, and other network layer protocols.
- **Classless Routing:** EIGRP supports Variable-Length Subnet Masking (VLSM), allowing for more efficient IP address allocation.

Next, let’s look at the top reasons why EIGRP is used as an IGP in large organizations:

1. **Scalability:** EIGRP can handle large and complex network topologies, making it suitable for large enterprise networks.
2. **Efficiency:** It uses incremental updates rather than periodic updates, reducing the amount of data that needs to be transmitted across the network.

3. **Flexibility:** EIGRP can be used for both IPv4 and IPv6 networks, and it supports multiple topologies and routing domains.
4. **Robustness:** EIGRP's DUAL algorithm ensures quick convergence and minimizes the impact of network changes, contributing to the stability and reliability of the network.
5. **Load Balancing:** EIGRP supports unequal-cost load balancing, allowing traffic to be distributed across multiple paths with different metrics.

EIGRP is very efficient; one of its key attributes is always having a feasible successor route instantly available, making it the fastest converging routing protocol. Using the concept of autonomous systems, networks can be built using mapping to BGP's autonomous systems. When the routes are distributed from one routing protocol to the other, EIGRP uses different administrative distances (ADs) for those routes so that it can trust what was learned through its internal peers within an autonomous system more than external networks.

EIGRP is one of the most common routing protocols to be used in all Cisco networks for small-to-medium sized businesses. Also, when it comes to dynamic multipoint virtual private networks (DMVPNs), this is the best choice because it scales well with small branch networks. One of the features of this protocol—Stub—becomes an ideal choice for remote branches because they are, in most cases, not transit sites.

With its fast convergence, BFD support, and also dual-stack capabilities, EIGRP is a suitable routing protocol for the underlay. A lot can be done with the EIGRP routing protocol; however, we will focus on three main components that make it suitable to be an underlay routing protocol and also highly resilient and scalable.

Redistribution

In a large network fabric-based network, redistribution of routes between routing protocols is inevitable. In the case of IGPs, the prefixes will be redistributed into and out of BGP at most times. (We will discuss BGP later in this chapter.) EIGRP has different metric values for redistribution. This helps with the trust factor, because routes that are learned from an EIGRP are always more trusted than the ones that are learned from other routing protocols. That makes sense because EIGRP will not have any control or influence in how other routes are treated in other protocols. This scalable design can be deployed right out of the box.

The administrative distance for internally learned routes within EIGRP is 90, and on the routing table, it is denoted as “D” routes. Any routes that are redistributed from other routing protocols have an administrative distance of 170 and are denoted as “D EX” in the routing table. Now, there is also a third administrative distance that is used for summarization, and that value is 5. Since summaries are noted internally, they are still denoted as “D” in the routing table. If we keep these hierarchy and administrative distances in mind, by default, EIGRP comes with a package that is highly scalable without the need for additional tuning. This also makes it highly resilient because all the out-of-the-box features ensure faster convergence and better troubleshooting with limited maintenance.

Routing Policies

Another piece of the puzzle in increasing efficiency is routing policies. Historically, EIGRP was a classful protocol. All the prefix advertisements, regardless of the size of the prefix, were advertised as classful. For example, if the subnet for a remote site is 10.1.1.0/24, when it is advertised out via EIGRP, by default, EIGRP would summarize it to 10.0.0.0/8 and advertise it across the WAN. This could create a massive issue because if there are 10 sites with 10.1.1.0/24 to 10.1.10.0/24 subnets, respectively, each site will advertise 10.0.0.0/8 by default. This would create routing issues with that massive summary from each site. To avoid this, the **no-summary** command was added to the routing instance to stop summarization and advertise specific prefixes.

Another bit of tuning to increase efficiency and resiliency of this routing protocol from an underlay perspective is to pair it with the BFD protocol so that it can converge faster and not have to wait for default routing protocol timers to detect a link as being down. BFD can detect a link failure in under a second. When a link is detected as being down, BFD will trigger an upper-layer IGP to inform its status, and IGP can quickly converge to a secondary route. This would work very well for EIGRP because it always has a feasible successor route ready to be deployed.

Scale Considerations

EIGRP can handle large-scale networks. Because it is a distance vector protocol, it does not need to know the full topology and perform the best path calculation based on that. This makes it an ideal choice for the underlay protocol. However, if there are non-Cisco switches or routers in the autonomous system, this protocol will not work. It is best suited for campus LANs because it has very high chances of a single-vendor environment and the IGP domain is also small.

OSPF

Open Shortest Path First (OSPF) is a link-state protocol. Different from the EIGRP, which is an advanced distance-vector, with link-state, OSPF has a bird’s-eye view of the network. OSPF uses the Dijkstra algorithm to calculate the best path for the destination prefix. Before making a decision on where to send the destination traffic, it calculates the path from an overall topology perspective. This capability makes it very versatile. Also, OSPF is an Internet Engineering Task Force (IETF) standards-based routing protocol, implying that it can work with any vendor’s networking equipment. In large enterprise networks, the networks are complex. In most cases, they have an organizational policy in place to not deploy proprietary systems, especially the ones making key decisions. The reason is to prevent the network from getting compromised or having to perform a massive overhaul in the event of lack of support. In this case, OSPF becomes a much easier choice to deploy.

The metric of OSPF is cost. The formula for cost is

$$Cost = \frac{Reference\ Bandwidth}{Interface\ Bandwidth}$$

When the OSPF routing protocol was drafted in the late 1980s, the maximum available bandwidth was significantly lower than what is available today. At that time, the maximum available bandwidth was 10 Mbps (10 Megabits per second) for Ethernet, 4 or 16 Mbps for token ring, and T1 lines were merely 1.544 Mbps. Hence, the reference bandwidth in the formula and all of the cost of the router were taken as 100 Mbps to future-proof the protocol cost calculation. However, today, where 100 Gbps (100 Gigabits per second) is normal, if we were to take the same old formula, the cost of 100 Mbps and 100 Gbps would result in just 1. Hence, to ensure the correct cost is configured and acknowledged across the network, today, it is recommended to change the reference bandwidth of all the OSPF routers to the highest capacity link in the network. This ensures that the correct cost is referenced across the network and that traffic takes the most optimal route.

OSPF is dual-stack capable, similar to EIGRP. However, unlike EIGRP, where both IPv4 and IPv6 routing processes run under the single EIGRP process, OSPF has to create a new process instance for each protocol stack. This brings up the discussion of resiliency and scalability concerns. OSPF networks employ the concept of *areas*, and all router links are part of an area. Due to OSPF's link-state nature, when a network event happens, every router that is part of a particular area needs to run the Shortest Path First (SPF) algorithm to ensure everyone has the correct outbound interface for their destination. This calculation becomes processor intensive with a network of hundreds of nodes and probably thousands of prefixes. Hence, architects divide the network into smaller areas that are typically mapped by regions or sites. All these smaller regions must be connected to Area 0, which is the backbone area. This ensures a proper hierarchy in the network design, and all the core node of an organization can be part of Area 0, connecting to different regions of the network.

To define a resilient network, a proper OSPF hierarchy is crucial. Similar to EIGRP, OSPF has a concept of a stub area that is usually configured for small stub sites that do not transit any through traffic. This design helps with better resource utilization and predictable convergence. It also prevents a site from accidentally becoming a transit site. Fine-tuning the OSPF network is critical because any misconfiguration could have a catastrophic effect. The OSPF protocol can be tuned in many ways to increase its efficiency for the part of the network for which it is configured.

Now, let’s look at the three fundamental parameters of OSPF and how can they be tuned for resiliency for underlay networks.

Redistribution

OSPF is a *classless protocol*. Hence, any network advertisement, by default, will advertise specific subnets. This is consistent with the entire OSPF routing domain, regardless of the OSPF area. Similar to EIGRP, OSPF has different route types. By default, all intra-area routes are denoted as “O” in the routing table. All inter-area routes are denoted as “O IA” routes. When the routes are redistributed from an external routing protocol into a normal area, they are denoted as “O E1” and “O E2.” By default, all redistributed routes are E2, and they have a fixed default cost of 20. Regardless of where that route is propagated, its cost will remain the same. E1 routes, on the other hand, have a variable metric. During redistribution, a seed metric is inserted, and every time that route is advertised further in the network, the cost of that link is added. This enables true cost across the network and is helpful when routes are propagated across the larger part of the network. There is also a special use case of stubby areas that are configured so that the particular area cannot become a transit for the traffic. When a redistribution is done in this area, it implies that an external network is connected to that area, and that area is “not so stubby.” Hence, similar to E1 and E2, those routes will show up as “O N1” and “O N2.”

Regardless of the route types of OSPF, the administrative distance of the routing protocol is 110. So the decision factor for the best path selection is not made based on the administrative distance, but rather by OSPF route type. In order of preference, the route selection is as follows:

1. O: Intra-area routes
2. O IA: Inter-area routes
3. O E1: OSPF External 1
4. O E2: OSPF External 2
5. O N1: OSPF NSSA (Not So Stubby Area) External 1
6. O N2: OSPF NSSA External 2

OSPF networks can grow very large and it is crucial to carry the true cost of the path across the network, so it is highly recommended to have a higher seed cost for the redistributed routes and ensure those route types are redistributed as type 1 (O E1). For small underlay networks where the hub-and-spoke topology is not being built, OSPF is widely used as the preferred protocol.

Routing Policies

Due to the link-state nature of the OSPF routing protocol, it needs to be aware of the entire OSPF topology of the network; it does not matter how big the network is or how many areas it has. One of the benefits of this protocol is that if a network event occurs in a different area, the full SPF calculation does not take place for that prefix. Since it knows what the exit points of the area are, it will do a partial SPF calculation and ensure that the path to that area border router (ABR) is accurate. This type of architecture also poses a larger issue of route summarization. Unlike EIGRP, where route summarization can be done on an interface level, OSPF route summarization can only be done at the area border routers. Consequently, designing the right size areas and having proper summarization are key in keeping the prefix churn to a minimum.

OSPF uses a concept of tags for its prefixes. It is highly recommended to tag the prefixes coming from an area, an external protocol, or a specific region so that if any route manipulation needs to be done, it can be done by matching the tag values. This is a highly versatile way to allow and/or deny prefixes from one area to the other, as well as from one routing protocol to another. This is a 32-bit field, and it is all numerical.

By default, in most of the routers, the reference bandwidth is set to 100 Mbps. If the network where OSPF is deployed has more than 100 Mbps links (today's networks will definitely have that), it is highly recommended to change the reference bandwidth for that OSPF domain to be higher to represent an accurate cost across the network. Also, be mindful of the fact that the gap between the lowest speed link and highest speed link is not too great, because the OSPF cost can go to a maximum of 65,535, and after that, the route prefix is deemed unreachable. For example, if a corporate LAN or data center has at least 100 Gbps links and if there is some remote site with a bandwidth of 1.544 Mbps (T1 line), changing the reference bandwidth for the entire OSPF domain to 100 Gbps will result in that remote site not being reachable because the formula, $\frac{100{,}000{,}000}{1500} = 66{,}666$, results in more than 65,535. This is an extreme scenario but highly likely if proper planning is not conducted while defining routing policies.

Lastly, BFD is another way to make the OSPF-based underlay network converge very fast. With subsecond detection, OSPF can converge in an instance and not rely on its default timers for the convergence. There is a way to converge OSPF in subseconds natively as well, but that method is too CPU intensive. Hence, BFD becomes a better choice in addressing that situation.

Scale Considerations

From a scale perspective, OSPF can be scaled for small-to-medium-size networks, but it is not recommended for large networks. The amount of SPF calculation that is done for all the prefixes and OSPF's nature to be aware of the entire topology does take some processor cycles. It is recommended that you have no more than 100 nodes in an OSPF area. OSPF is useful for a campus location but needs to have a BGP-like protocol that can be redistributed in to break that large domain and reduce the prefix churn. All the newer fabric-based architectures follow similar principles for having a good-size scale right out of the box.

With an increase in interface bandwidth and 100 Gbps being the norm today, fine-tuning the reference bandwidth becomes crucial. Being able to upgrade an old OSPF-based network to newer standards requires highly methodical planning.

IS-IS

The Intermediate System to Intermediate System (IS-IS) routing protocol is also a link-state protocol similar to OSPF but is highly versatile and scalable. It is a *connectionless* network service protocol, which means it was designed to route not only IP but also non-IP traffic. There is no dependency of IP addressing when this routing protocol is configured, so it can route and also dampen the transport prefixes from advertisements. IS-IS is similar to OSPF: Both are link-state protocols, need to be aware of the entire topology to make routing decisions, and employ the concept of areas. IS-IS, however, has a much less complicated area structure with only two types of areas: Level 1 or Level 2, with Level 2 being the higher or backbone area.

Instead of links being in an area like OSPF, in IS-IS, the entire router is an area. By default, a router is configured as both Level 1 and Level 2. It can be specified manually if either area is desired, and it is recommended that you select one area for better efficiency. IS-IS requires less configuration compared to other routing protocols. Due it its connectionless nature and low maintenance, it is highly desirable in service provider networks. Almost all of the service providers have IS-IS as their IGP routing protocol, with MPLS VPN, and BGP running on top of it. IS-IS has many inherent capabilities, such as setting a maximum overload bit, which will enable a router to advertise all of the prefixes with the highest metric. This is a graceful way of taking a router out of the majority of traffic forwarding so that network maintenance can be carried out easily.

Historically, this protocol is not very well known because it was mainly deployed on service provider networks, and this protocol suite was not available on standard IOS images. Enterprises resorted to EIGRP or OSPF as their routing protocol of choice because they are easily understandable and do not require major tuning right out of the box. Today, with controller-driven architectures, IS-IS is making its way into enterprise networks because most of the configuration and tuning are handled by network controllers such as APIC and Catalyst Center. So, by default, technologies such as Application Centric Infrastructure (ACI) and Software-Defined Access (SD-Access) build their fabric architectures with IS-IS as their underlying routing protocol.

Because today’s networks and, most importantly, fabric-based networks have identical links across the entire infrastructure due to their use of equal-cost multi-path (ECMP), IS-IS works out to be the best protocol. The reason is that, by default, it uses default cost for each interface. There is no calculation of a bandwidth or a complex formula. Each interface is given a cost, and a route to the destination is summation of all the interface costs in its path. This approach is highly simplified and especially good when the underlying infrastructure uses similar capacity and a proper hierarchy. Also, with capabilities such as its connectionless network service, it does not need an IP address to form a peering relationship with other routers. This makes it robust and efficient with a single- or dual-stack operation working in tandem.

Next, let’s look at three main categories to understand how this protocol can be most efficient and robust for underlay networks.

Redistribution

In the IS-IS routing protocol, there are two types of redistribution: internal and external. Internal redistribution occurs between different IS-IS areas. Level 1 areas are considered to be more stub in nature, whereas the level 2 area is considered as a backbone. You should always ensure all areas are connected to the level 2 area and that the level 2 area is contiguous. Any break in that would result in undesired IS-IS topology. By default, all Cisco routers are programmed to run both IS-IS areas under the IS-IS routing process. This means that each IS-IS router will establish two peering relationships with adjacent routers: on level 1 and level 2 areas. This approach is inefficient because running two different subprocesses for the same topology is not ideal. Hence, it is highly necessary to make sure all the routers are configured with the proper IS-IS hierarchy and that the routing process be changed to address and reflect the correct area level. For the area border routers (ABRs), they must take part in both areas because they need to peer with IS-IS level 1 routers on the stub side and IS-IS level

2 on the backbone side. All the backbone routers, unless they are ABRs, should be configured only as level 2 routers.

In the normal IS-IS topology, level 1 routers have access only to prefixes in their own area and default routes to the other areas and backbone. On the other hand, level 2 routers, since they are backbone, have all routing prefixes because they connect all the other areas. This is natural because to increase efficiency, you do not want to advertise all the prefixes in the stub area if their exit is only two ABRs. So, by default, IS-IS does a good job in maintaining the proper hierarchy and ensuring the right prefixes go in the right area.

There are two main use cases where you would want to advertise nonstandard prefixes to either levels and deviate from default settings:

1. **Level 1 to Level 2:** By default, all level 1 route prefixes are injected into level 2. This is good from an SPF calculation perspective because it does not do full calculation in terms of prefix flap; however, all those prefixes are still in the Routing Information Base (RIB). If the IP addressing follows good standards, it is advisable to summarize the prefixes going to the level 2 area from level 1 to reduce the routing table entries.
2. **Level 2 to Level 1:** By default, only a default route is allowed in to a level 1 area from the ABR. There might be a requirement for any specific application or overlay tunnel that may need a specific IP subnet or a /32 loopback IP to form that tunnel. In this case, at the ABR, a route-map with that specific prefix can be advertised into level 1 besides the default route to achieve such a purpose.

External redistribution is standard when it comes to IS-IS. It uses the seed metric from any protocol, and from that point onward, it calculates all the links to the destination because that prefix goes into the IS-IS routing domain. It is advisable to use some form of route-map to control that prefix advertisement for better security, even if it comes from a trusted network. If a major event occurs, that route-map provides an instant place to add problematic prefixes in the prefix-list and quickly block them. There are no different administrative distances or metrics for external protocols into IS-IS.

Routing Policies

The IS-IS routing protocol, similar to OSPF, needs to be aware of the entire topology. Despite the fail-safes built into the protocol, a few considerations must be considered. A few features can be enabled to improve efficiency and resiliency on this protocol:

- **Route Aggregation:** As discussed in section on IS-IS redistribution, route aggregation can be configured on the ABRs to reduce the routing table size and also to improve convergence and performance. For this feature to work effectively, you need to ensure that the IP address design and hierarchy are done properly. This way, you avoid the risk of a prefix being black-holed by a route summary and losing network connectivity or pointing to a wrong part of the network.
- **Route Filtering:** Similar to route aggregation, route filtering can also increase efficiency. This efficiency is in terms of the number of prefixes that are reduced from one domain. Route filtering helps with fewer SPF calculations in the event of prefix flaps and also conserved memory.
- **Topology Hiding:** IS-IS employs a neat feature to not advertise connected or transport links. This massive feature can provide huge benefits. Essentially, because point-to-point connections between routers or switches in the underlay just help with movement of packets and not a destination, IS-IS can hide them. This means that the main IP addresses, such as loopbacks and any other management IPs, are advertised normally. In a scenario with 50 switches connected to 2 distribution switches, a total of 100 links and point-to-point IP addresses are configured, and these are usually /30 or /31.
- **Tags:** IS-IS tags are another key element that can be attached to a specific set of prefixes as they move around in the network. When you use these tags, it becomes easier to apply routing policies across different IS-IS routers.

Scale Considerations

The IS-IS routing protocol provides massive scale advantages. Due to its efficient nature and simplified configuration, it comes with significant fine-tuning right out of the box. With OSPF, the recommendation is to have about 100 routers in an OSPF domain, whereas with IS-IS, this can be thousands. Hence, IS-IS is a protocol of choice for service providers due to their span nationally and, in many cases, globally. This protocol can scale and provide a robust backbone for the MPLS network.

Overlay Routing Protocols

In the preceding sections, we looked at the resiliency and effectiveness of underlay routing protocols and how they build a solid foundation of any network infrastructure. Underlay routing allows subsecond convergence and ease in taking down a device for maintenance, whereas overlay routing offers a feature-rich fabric that can be isolated to provide additional services on demand. With a combination of these two protocols, this is a dream come true for any IT management team that can provide business continuity with the latest on-demand features.

With traditional networks, the control plane, data plane, policy plane, and management plane are almost monolithic in nature. Each device is configured and managed individually to make sure all of these objectives are achieved. The moment that you need macrosegmentation, you need to come up with separate routing planes on all devices that are simply not scalable across the hundreds of devices in the network. Additionally, this increases processing on the devices because multiple instances of the same routing protocol might be needed, and one of the most important issues is the downtime of the network when a device needs to be taken down for maintenance.

The overlay routing protocol works on the fundamental principle of *packet encapsulation*. Essentially, to route a packet over a routed infrastructure, you need to encapsulate a packet in such a way that it is transparent to the underlying network until it reaches its destination. For that to happen, a packet basically needs to be “tunneled” from its source to its destination, and the source and destination devices need to be fully capable of encapsulating and decapsulating the packet effectively. Figure 8-2 illustrates the packet encapsulation for IPsec. The basic idea here is to tunnel the traffic from the source to the destination without any device in the middle being aware of the original traffic. This feature has been used for decades, such as the IPsec VPN and GRE tunnel. These are nothing but individual point-to-point tunnels that are configured between a set source to the set destination, and any traffic between the devices is tunneled in.

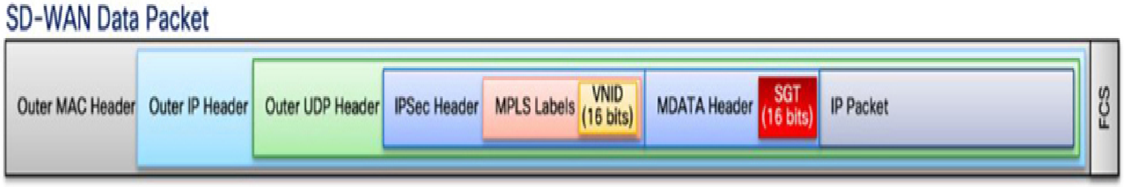


Figure 8-2 *Encapsulation of SD-WAN Data Packet*

Despite the traditional VPN technologies providing encapsulation and decapsulation over a large network, these point-to-point tunnels are not scalable, and also if they are not configured properly, they can have massive consequences in terms of network routing and convergence. Organizations need to develop a proper thought process to have them scale and be more effective in configuration management. Hence, when overlay benefits needed to be leveraged for large-scale networks that needed isolation and stretching across various devices, this overlay concept needs to be broken down into four major parts: the control plane, data plane, policy plane, and management plane. Each plane provides a set amount of control and scalability that can be used in conjunction with the underlay networks.

Now, let’s look at some overlay routing protocols and see how they provide the scalability and resiliency that is required. We will start with some of the oldest protocols and move toward some modern protocols. All these protocols will be compared against all four planes—control, data, policy, and management. The technologies mentioned in subsequent sections are so vast on their own that multiple books have been written on each of them. Hence, for the sake of this high-level discussion, we will take a 100,000-foot view of these technologies.

MPLS-VPN

Knowledge of the term *Multiprotocol Label Switching—Virtual Private Network (MPLS-VPN)—MPLS* for short—in the networking industry is as common as breathing. MPLS was formed on the basic principles of overlay networking to ensure that macrosegmentation is contained within a VPN and also stretched across geographical locations. Its capabilities include providing faster convergence, inserting on-demand service, and leveraging advanced features of routing protocols. Today's service providers not only use MPLS-VPN for Layer 3 routing and segmentation but also for other layers. With services like the Virtual Private LAN Service (VPLS), pseudowire, and effective optical layer convergences, MPLS has become a feature-rich technology that can not only scale but will last for years to come. Let's look at its four important aspects.

Control Plane

The control plane of the MPLS-VPN network is built using Border Gateway Protocol. With advanced BGP policies and attributes, macrosegmentation is configured to provide routing table separation and extension capabilities. BGP's multi-address family capability provides per-VRF routing segregation and also redistribution of customer-facing routing protocols into a single instance of the routing protocol on the service provider side. One of the main reasons for MPLS being successful is that, regardless of what routing protocol an organization is running in its network, the service provider can always take that routing protocol and redistribute it into BGP. This provides simplicity and ease of management for the entire infrastructure. The edge routers on the service provider network known as provider edge (PE) have the most complex configuration because it connects with many service providers customers and could potentially have tens or hundreds of VRFs. All of that macrosegmentation is handled by BGP.

Today, because BGP accessibility is common across enterprise networks, most customers prefer to have their service provider peering via BGP than any other routing protocol. This further simplifies the network on the service provider side and gives proper network control on the end consumer side.

Data Plane

The data plane of the MPLS VPN networks is built by the Label Distribution Protocol (LDP). LDP is considered a Layer 2.5 protocol because it does not have an attribute of Layer 2 or Layer 3 protocols. Essentially, MPLS has a label that is like a shim that is inserted in the IP packet and is transported across the vast service provider network. These labels are numerical values that identify a specific prefix and/or customer network. BGP, in conjunction with LDP, takes these labels to the next level and basically creates a multilabel packet that is transported across the entire network. The outer label provides per-hop transport to the intended destination, and the inner label provides identification of the end VPN, which is used by a remote PE router to know which VRF this packet belongs to. Using labels for packet forwarding eliminates service providers running BGP in their backbone to increase efficiency, performance, and convergence. Label convergence is very effective, and with the help of the fast reroute (FRR) feature, a stand-by route can be ready to go in the event of a node or a link failure. Service providers charge an additional premium to have link or node protection for their customers and increase their SLAs.

Policy Plane

For the policy plane in MPLS-VPN networks, the BGP routing protocol is used. Since MPLS-VPN is a transport protocol and no actual end users connect to it, there is no need to have a high level of endpoint-based security deployment. All traffic coming in to the MPLS network is transit, so the policies in this case are purely to ensure that one customer’s routers and prefixes are not leaked into another customer’s network. These policies are maintained and audited on a regular basis to ensure data isolation is maintained.

Management Plane

MPLS-VPN has been around for decades. With the evolution of the protocol’s features, the management of this protocol has also changed to more of an automation-driven approach. Since service providers have more than one router type and vendor, usually orchestration tools such as Ansible and Terraform are used to build automated networks for their customers and manage the network. This method ensures serviceability and, most importantly, reduces the human-error element that can take down a network. Most service providers have their own scripts that take the addition of a new VRF or a circuit for a customer and automatically provision it across the network.

BGP-EVPN

Border Gateway Protocol—Ethernet Virtual Private Network (BGP-EVPN) is a powerful overlay protocol that is designed for data center environments. Data centers have unique requirements compared to service provider IP transport networks. Besides providing transport to the different applications, BGP-EVPN needs a low-latency network and also a stretch fabric. The virtual servers are connected to the network, and they also need to be moved from one physical server to another for maintenance. A major Layer 2 connectivity requirement needs to be met. With data centers spread into racks and rows, today's modern data center architecture follows the CLOS architecture—spine and leaf. The CLOS architecture provides the utmost scalability, with spines being the backbone that connects all the leafs, which are top-of-rack (TOR) switches. All servers connect to leafs, and each leaf provides one hop connectivity to another leaf.

As the name suggests, Ethernet Virtual Private Network is an extension of the Layer 2 network across the data center. Despite BGP-EVPN providing very efficient Layer 2 services across the data center, it also has feature-rich Layer 3 services that are able to extend outside of the data center as well. With its multitenancy capability, the data center can be segmented into small virtual networks or tenants, and any traffic that is leaving outside of the data center can take the Layer 3 network handoff.

Recently, BGP-EVPN has evolved into enterprise campus networks as well. This solution is compatible with campus switches as well, so for many organizations, especially where they have an on-premises data center, extending that BGP-EVPN fabric into the campus becomes a much better choice. This provides a natural extension of their data center macrosegmentation into the campus.

One of the major differences between MPLS and BGP-EVPN is the ability of the BGP-EVPN to provide microsegmentation as well. The data plane mechanism used in this solution has the ability to carry macro- and microsegmentation information within the data packet itself. This capability is crucial because in this fabric, unlike MPLS, hundreds of endpoints are connected instead of routers transporting traffic. These endpoints are in the form of servers or applications in the data center or end-user devices if BGP-EVPN is chosen to be deployed in the campus network.

Next, let’s look at the four main components of this solution.

Control Plane

The control plane of BGP-EVPN is provided by the BGP routing protocol. All the devices in the network need to run BGP—from the border leafs to the edge leafs. The BGP routing protocol uses the L2VPN EVPN address-family and builds the peering relationships with all the BGP running routers. Since the nature of the BGP is to have full-mesh peers, for large networks, the spines in the network run the route-reflector feature to limit the number of BGP peerings between each of the devices. This improves the performance significantly.

BGP is a memory-intensive protocol because it is designed to store a large number of prefixes in the routing table. In this solution, BGP needs to be aware of all the prefixes that are in the overlay so that it can provide the most direct and shortest routing. The BGP-EVPN uses a push model for its control plane. This means that each switch needs to be aware of every single prefix in the overlay virtual network. Any changes, such as addition or removal of an endpoint, will be pushed to the entire fabric and all of the switches to ensure they are all aware of the full topology. This is not a big problem when BGP-EVPN is used in the data center, because in most cases, there is not too much churn in terms of prefix moves from servers. They are mainly set and usually move only when they are under maintenance. However, when it comes to the campus side, this may create a bigger scale issue when there are hundreds of switches and thousands of users who are constantly roaming across the network. Constant BGP updates from roaming users could cause a bigger churn in the network with this push model.

One of the other notable concerns with BGP-EVPN design is the recommended use of identical switch models across the entire fabric. This is not a problem for the data center fabric because all the switches used are always identical in performance. However, when it comes to campus networks, this could pose a problem because not all switches are identical on the access layer due to different port requirements. Some IDFs may require stacks of switches or a chassis-based switch, whereas in some, it might be just a single switch to provide connectivity for a few endpoints. Having a high-performance switch to accommodate a large BGP table and to connect a few endpoints might not be ideal in terms of cost. Hence, this is a great solution for data center[nd]based fabrics, but it has its limitations on the campus side of things.

Data Plane

To carry macro- and microsegmentation information across the network, BGP-EVPN uses virtual extensible LAN (VXLAN). VXLAN is an encapsulation mechanism that extends Layer 2 networks over a Layer 3 infrastructure. As shown in Figure 8-3, VXLAN encapsulates Ethernet frames within UDP packets, enabling them to be transmitted over a Layer 3 routed network. This encapsulation uses a 24-bit segment identifier known as the VXLAN network identifier (VNI) allowing up to 16 million unique identifiers compared to the limit of 4096 VLANs in a traditional network.

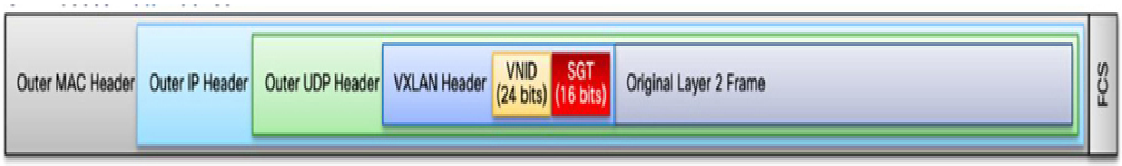


Figure 8-3 *VXLAN Encapsulated Packet*

VXLAN encapsulation and decapsulation are performed by the leaf switches in the fabric, which are also known as VXLAN tunnel endpoints (VTEPs). VXLAN brings scale, isolation, and network resiliency into modern networks. VXLAN is not only limited to BGP-EVPN, but the same data-plane mechanism is used in technologies such as Application Centric Infrastructure (ACI) and Software-Defined Access (SD-Access).

Transport of critical information, such as macro- and microsegmentation, over a data packet is highly efficient instead of relaying that information through control plane mechanisms. With control plane-based information, the device's processing power is reduced as it increases the load on the processor. When the same information is embedded in the data plane, it reduces the processor overload, and the critical information is passed all the way through the transport network to the end device, which makes the final decision on decapsulation and forwarding the original packet to the end device based on the applicable policies. This method, despite reducing the load on the overall transport network, has one drawback: Since the final decision regarding the packet forwarding is made on the final leaf connected to the endpoint, the flow that needs to be dropped will not be dropped until it reaches the destination. Policy enforcement is at the egress instead of ingress of the fabric.

Policy Plane

The policy plane on this solution can leverage all the benefits of VXLAN encapsulation. However, because the control plane is BGP based, Layer 3 policies are controlled via the BGP routing protocol, and all microsegmentation policies can be controlled via security group tags (SGTs) at the leaf layer. Cisco TrustSec can be leveraged to conduct microsegmentation across the network, but, by default, it is not used in many BGP-EVPN deployments. The following mechanisms are used in the policy plane for BGP-EVPN.

- **Route Policies:** These policies use a combination of route-maps, prefix-lists, and community attributes that are used in BGP for route manipulation.
- **Control Plane Mechanisms:** BGP features such as route reflectors and confederations are used for better propagation of prefixes. BGP attributes can be used for path preference.
- **Network Segmentation:** VRFs and VNIs are used for routing plane separation. Having different routing planes for different business units, tenants, or functional parts of the networks is vital for leveraging the same underlying network infrastructure for scalability and efficiency.
- **Security Policies:** Network access control lists (ACLs) are used to allow or deny traffic in and out of an interface.
- **Automation and Orchestration:** Various network automation tools can be used to dynamically add, modify, or remove policies based on triggered events. This can help with scale and eliminate some management overhead.

Management Plane

BGP-EVPN is not a controller-driven technology. Some simple rules need to be followed when BGP-EVPN needs to be configured, and these rules can be automated using custom scripts or workflows. Hence, the management plane of BGP-EVPN can be deployed in the following ways:

1. Manually using configuration commands on all the switches
2. Automatically by leveraging Ansible playbooks, Terraform providers, or custom scripts
3. Controller specific to device models, such as Nexus Dashboard Fabric Controller NDFC) for Nexus switches and Cisco Catalyst Center for Catalyst 9000 series switches

Using these mechanisms, you can configure and manage BGP-EVPN in an organization's network.

SD-Access

Cisco SD-Access is a fabric-based architecture designed for campus networks. It is a highly scalable, fully orchestrated, and intent-based networking solution. SD-Access is configured and managed by Cisco Catalyst Center and is designed to be run and operate on Cisco's Catalyst 9000 series switches, wireless LAN controllers, and access points. Although Cisco Catalyst Center can manage a variety of platforms, for this section, we will focus on the SD-Access component.

SD-Access has similar characteristics to BGP-EVPN in terms of fabric-based architecture. It is designed to be more efficient and addresses some of the major drawbacks that are part of BGP-EVPN and with the additional benefits of an integrated wireless architecture. It uses Locator ID Separation Protocol (LISP) instead of BGP.

Now, let's look at its four main components to see how it is an effective solution for campus networks.

Control Plane

In SD-Access, control plane functionality is provided by LISP, which is a highly scalable protocol and also the IETF standard defined in RFC 6830. LISP's scalability comes from the protocol's capability to reduce the size of the routing table by separating IP addresses into endpoint identifiers and routing locators. LISP employs the concept of *map server and map resolver (MS/MR)*. These components can be a single device or separate devices that can provide scalability. MS is responsible for receiving and storing the endpoint ID (EID) to routing locator (RLOC) mappings, and MR is responsible for answering the queries from the routers and providing them with the mapping entries. For SD-Access, MS/MR functionality is combined in control plane nodes of the solution. Basically, this functions like a Domain Name System (DNS). Just like a computer queries a DNS server for the location of an IP address of a URL or system name, when a router or switch part of the LISP network needs to reach a destination, it queries an MR, which in turn queries the appropriate map server to find the mapping of the end device (EID) and where it is located—the RLOC. Once the router or a switch receives that mapping information, it will then build a LISP tunnel, where an original packet is encapsulated into a LISP header and sent to the destination router. This is the default and native behavior of the LISP routing protocol.

In the case of SD-Access, LISP is used only for the control plane function, and its data plane is not used at all. This modification is what makes SD-Access a viable solution for campus networks. Unlike BGP-EVPN, which is a push model, where all devices need to be aware of all the prefixes at all times, LISP works on a pull model. Here, end devices only need to know about the prefixes they need to communicate to. Hence, it becomes more efficient because the entire routing table does not need to be downloaded onto the end device. This overcomes a major hurdle in the BGP-EVPN fabric in the campus, where there can be multiple switches with different performance levels in the fabric, and they will cache only in the prefixes they need to communicate to, with an ability to time out some entries when they are not used for a long time.

For an SD-Access solution, LISP has been tweaked a little bit to allow better handling of roaming and mobility. When it comes to wireless networks, LISP becomes very effective because it provides an ability to decouple the control and data planes of the wireless traffic. All the wireless control traffic traverses via the CAPWAP tunnel to the wireless LAN controller, and all the data traffic gets offloaded directly from the access point to the edge node. This is a massive boost in terms of performance because high-speed and high-capacity data traffic now gets directly offloaded to the switch instead of being backhauled to the WLC. Wireless essentially becomes an extension of the wired network, where it can even share the same subnet between wired and wireless endpoints if required.

Data Plane

The data plane of SD-Access is exactly the same as that of BGP-EVPN. It uses VXLAN encapsulation to transmit data from one edge node to the other. We covered benefits of VXLAN in the “BGP-EVPN” section; please note that the benefits are identical with an exception on how security and wireless work.

We know that VXLAN can embed macrosegmentation in its packet header. This is covered by VNID. In SD-Access, Cisco TrustSec provides a fully integrated microsegmentation strategy that can scale across an entire campus fabric. With integration of Identity Services Engine (ISE) in this solution, dot1x can be enabled on every switch port of the edge node, and by leveraging Cisco’s TrustSec, a security group tag can be assigned to that endpoint. This SGT can be propagated in the VXLAN tag along with the VNID to have a unique identity for that endpoint across the network. This is a crucial point to consider because SD-Access is designed for campus networks where a multitude of endpoints constantly roam and have different signatures. Having network access control (NAC) embedded in the solution makes the network really secure. Once the endpoints are classified and tagged in the SD-Access fabric, by default, they are free to communicate and roam in the same VRF. However, when the TrustSec matrix is configured by creating secure group access control lists (SGACLs), each endpoint can be allowed or denied communication with other endpoints even if they are in same VLAN or same switch. This security mechanism takes enhancing security in the network a step further. Here, endpoints (printers, badge readers, sensors, and so on) can all effectively be placed in a single VLAN, and using SGACLs, direct communication between these types of devices can be blocked. An SGACL can be configured so that printers can communicate only with print servers, badge readers with the security system, and sensors to the building management system. The possibilities are endless.

When it comes to wireless, all the fabric-enabled access points build a Layer 2 VXLAN tunnel with the upstream switch so that they can send all the traffic from a particular VLAN to that switch under VXLAN encapsulation. Once the packet reaches the edge node, it will decapsulate it, perform any policy application, and then decide if the packet needs to be dropped or needs to be forwarded to a local or remote location using VXLAN encapsulation. When it comes to roaming, when an endpoint moves from one access point to another, the switch sees WLC, and the switch will communicate with the control plane of the SD-Access fabric to update their EID-RLOC mapping to another switch. This move is seamless, and traffic is built and tunneled across the network more seamlessly.

Policy Plane

The policy plane for SD-Access is Cisco TrustSec. ISE is a major component of the solution to provide complete secure design. ISE is not mandatory to use the SD-Access fabric. Any NAC solution can be used, but that will not allow you to use all the microsegmentation features of SD-Access. Microsegmentation in SD-Access means using TrustSec. All of the security policies can be controlled via ISE, where authorization rules that classify the endpoints are built. These endpoints can be profiled and postured, matched with a specific condition such as device type or operating system, and then assigned a VLAN and SGT. ISE passes the final authorization information to the switch, allowing the endpoint to access the network based on the defined policy. These policies are centralized, which means they can be created once and can be applied to all SD-Access[nd]enabled sites.

The TrustSec matrix is passed on to all the switches by ISE as well. These SGACL policies are dynamic and instantaneous, not part of the switch configuration. When these policies are created, they take effect immediately and help with networkwide control of communication between different devices.

The SGACL policies can also be propagated to Cisco's Firepower Threat Defense (FTD) firewalls and Palo Alto firewalls. These next-generation firewalls are able to understand VXLAN tagging and can look into the packet header to identify their SGTs and enforce policies based on the TrustSec matrix.

Management Plane

The management plane of the SD-Access solution is provided by Cisco Catalyst Center. This is a single plane of glass that contains comprehensive workflows to design, provision, and manage one or more SD-Access sites. This management plane does not require detailed network configuration as if you were to build a running configuration of all individual switches in the network. You need to specify IP pools and intend to deploy them with the right security policy, and Catalyst Center will build the configuration for all the switches and routers in the fabric site and deploy them automatically. Catalyst Center can also be used for management of all the switches in terms of software upgrades and troubleshooting.

An additional benefit of Catalyst Center is providing assurance of all network devices, endpoints, and traffic. It has Application 360, Device 360, and User 360, which provide comprehensive statistics. In Application 360, Catalyst Center indicates how each application behaves in the network and its performance within the fabric. In Device 360, it provides statistics about the device and users connected to the device, along with the type of traffic flowing through the device. With User 360, it gives a fingerprint of all the users, indicating how many devices they have in the network and what types of applications they are accessing and their performance. This solution provides a single pane of glass for campus networks that can be scaled within a region or across an enterprise.

ACI

Cisco Application Centric Infrastructure (ACI) is a comprehensive fabric-based data center network solution. ACI is an alternative to the BGP-EVPN-based solution, addressing many drawbacks. The goal of ACI is to simplify, optimize, and accelerate the deployment of applications through a comprehensive framework of automating network resources. ACI integrates software and hardware effectively to provide a scalable solution or data center network. It is highly modular and driven by a centralized policy framework. The centralized policy defines and enforces set business rules across the network, ensuring consistent application performance and security.

Application Policy Infrastructure Controller and all the underlying switches work in unison to drive application performance in the data center. Similar to BGP-EVPN, ACI uses a CLOS framework of spines and leafs to build the underlying infrastructure. The IS-IS routing protocol is the underlay routing protocol of choice.

Control Plane

ACI uses Council of Oracles Protocol (COOP) to communicate mapping information, such as location and identity to the spine proxy. This is, in function, similar to LISP in SD-Access. Here, instead of a dedicated control plane node like in SD-Access, ACI spine switches are considered a source of truth. A leaf switch forwards endpoint address information to the spine switch, which is referred to as “Oracle,” using a zero message queue (ZMQ). One of the major roles for COOP is to ensure that spines maintain a consistent copy of endpoint addresses and location information. It also maintains a distributed hash table (DHT) repository of endpoint identity to a location mapping database. From a security standpoint, COOP data path communication prioritizes secured connections, and it uses the MD5 option to protect messages from malicious traffic injection.

With COOP ensuring that all endpoint data is consistent across all spines, giving high priority and secure communication to its data path, and providing MD5 authentication support, it becomes highly scalable and modular. This can create a complexity in management; however, all the configuration and management workload is handled by the APIC, making it easier to use.

Data Plane

Similar to BGP-EVPN and SD-Access, ACI also uses VXLAN encapsulation in its data plane. We have already covered the benefits of VXLAN from a macrosegmentation perspective. From a microsegmentation perspective, VXLAN provides the endpoint group (EPG) capability in ACI. EPG is a logical grouping of hosts known as endpoints that have common policy requirements, such as security, virtual machine mobility (VMM), QoS, or Layer 4 to Layer 7 services. Similar to SGTs in SD-Access, EPGs enable you to manage endpoints as a group rather than individually manage them. This capability improves scale and performance. Each EPG belongs to a bridge domain (BD), and traffic communication between the same EPG is not blocked by default. Any communication between different EPGs is blocked by default from a security perspective. This is where contracts are created to allow specific traffic between different EPGs.

ACI's EPG and SD-Access's SGT use the same segment ID bit in the VXLAN header. This makes connecting the two fabrics easier. ISE can be used as a moderator to keep track of SGTs and EPGs. With SD-Access and ACI integration, the traffic is passed from one domain to another using a VXLAN data packet, and contracts can be created in their respective domains to allow or deny communication between them.

Policy Plane

The largest component of the policy plane in ACI is contracts. These contracts are used to allow or deny communication between different EPGs. These contracts usually follow business justification on what type of communication is allowed between the EPGs. These policies are deployed via APIC in a centralized fashion.

At a very high level, the policy plane abstracts complex network configurations into high-level policies that reflect business and application requirements. This abstraction simplifies network management by allowing administrators to focus on what a network should achieve instead of how it should be configured.

Management Plane

Configuration of ACI in the data center is handled via an APIC controller. The controller and switch configuration can be fine-tuned to be configured with zero-touch. When the set of switches is booted up, the APIC controller discovers the devices and configures all relevant underlay and overlay configuration automatically. Today, ACI has evolved to a stage where APIs are used to configure entire data centers via automation. APIC is highly API capable. With automation and large-scale device onboarding to granular policies, everything can be configured via APIs. The user interface is robust and can be easily integrated into an organization's CI/CD pipeline. All the work from building the fabric to automated upgrades can be taken care of by the APIC controller and can be automated by leveraging their APIs.

SD-WAN

Cisco Software-Defined Wide Area Network (SD-WAN) is a fabric-based solution for the WAN domain. The WAN was one of the first domains to embrace overlay technologies. With service providers providing MPLS as a standard, when Internet circuits became cheaper and with higher bandwidth, many small-to-medium-size businesses started to leverage the Internet for their site-to-site communications. Since the Internet was not secure and NAT blocked direct client-to-application access, the IPsec VPN was used to build site-to-site point-to-point tunnels. This provided a simple but very effective way to connect a few remote sites. For large organizations, they needed diversity. They needed SLAs and the stability of private MPLS-based circuits, but when they started to have hundreds to thousands of locations in their network, maintaining redundancy with dual MPLS connections became very expensive. So, they opted to utilize one Internet-based circuit as a backup or employ traffic engineering to use Internet circuits for bulk and less mission-critical traffic. This method started to prove more effective, but scale and management were still issues. This is when dynamic multipoint virtual private networks (DMVPNs) were introduced. DMVPNs used multipoint generic routing encapsulation (GRE) tunnels to establish connectivity between the sites, essentially giving MPLS-like network connectivity over Internet circuits.

With DMVPN providing MPLS-like backup connectivity, a more advanced software-driven solution named Intelligent WAN (I-WAN) was developed by Cisco. I-WAN was able to leverage DMVPN with performance routing (PFR) to identify traffic type and intelligently route over either MPLS or Internet circuits instead of relying on simple source or destination prefixes. This solution lacked an automated control plane until Cisco acquired a company called Viptela that provided a more robust solution with a centralized controller to deploy software-driven WAN. Today, SD-WAN is widely used in enterprise networks to provide scalable WAN fabric over any kind of transport. It is transport agnostic. Whether a transport is MPLS, Internet, wireless, LTE, or 5G, it does not matter. SD-WAN will form dynamic tunnels and give secure connectivity to all the sites.

Next, let’s look at the components of SD-WAN and understand its inner-working at a high level.

Control Plane

SD-WAN’s control plane is built using the Overlay Management Protocol (OMP). This proprietary routing protocol is similar to BGP. There is a good similarity between SD-WAN and SD-Access in terms of logical operation. Similar to the SD-Access control plane nodes, SD-WAN has Catalyst Controller (formerly vSmart) to provide control plane functionality. In this case, Catalyst Controller functions as a route reflector to ensure all the edge routers have the right centralized and local policies deployed on them. Having central control and policy enforcement is crucial to securing the WAN and provide the utmost scalability.

This control plane is first formed by connecting all the transports to SD-WAN and establishing dynamic secure GRE tunnels over each of the WAN transports. Once the WAN edge router is discovered and onboarded to the fabric, these tunnels are formed, and based on the transport preferences, routing policies are injected in the routing table, preferring one transport over the other for each type of traffic. On top of this, SD-WAN has multitenancy; therefore, multiple business units or functions can be configured with their own unique policy over the same WAN. For example, in a typical corporate network, there is a trusted corporate network and an untrusted guest network. With SD-WAN, they can both be in separate VRFs with all trusted corporate traffic following full-mesh network connectivity to each site, and all guest Internet traffic can be configured to be hub-and-spoke to the central facility where it can be inspected and exited out of the network.

For an SD-WAN to work effectively, control connections to the Catalyst Controllers and Catalyst Manager need to be fully operational. That is the heartbeat that keeps the WAN up and operational. WAN circuits are known to have outages and issues because they are outside of the organization’s control. So, in the event these circuits are down, WAN routers do not stop forwarding or change policies; instead, they cache in some of these policies for a predefined time and ensure traffic gets forwarded via other available circuits as long as possible.

Data Plane

When it comes to data plane traffic, SD-WAN encapsulation uses MPLS labels’ VPN-IDs to map VRFs. With newer enhancements, the microsegmentation information is transmitted in MDATA header’s DATA field. This enables SD-WAN to successfully integrate with SD-Access and provide end-to-end identity and security across the WAN. The encapsulation and decapsulation functions occur between two WAN edge routers. One of the biggest benefits of SD-WAN over solutions like DMVPN or IPsec is that it does not need to create multiple tunnels or routing instances over the same transport for different VRFs. If an organization had five VRFs for their business function, with the use of DMVPN or IPsec, there would be five DMVPN tunnels or IPsec VPN tunnels per router to multiple sites to maintain that macrosegmentation. Whereas in SD-WAN, there is only a single tunnel and all the VRF-based traffic is isolated using MPLS headers and transported over the same tunnel across to any site defined in the routing policy.

Policy Plane

The policy plane of SD-WAN is divided into two major parts: the routing policy and data policy. Since SD-WAN is a transport network, similar to MPLS, no direct end users are connected to the network. Its primary function is to provide inter-site connectivity and transport data as efficiently as possible.

With the routing policy, the flow of information in the control plane of the SD-WAN architecture is determined. This examines the routes that are learned and how they need to be distributed across the network. This control policy is unidirectional and can be deployed at a list of sites in an inbound or outbound direction that impacts how the traffic flow will be configured. The routing policy is crucial because the data policy relies on an effective routing policy and how efficiently traffic moves across the SD-

WAN domain. The data policy directs which type of traffic takes which path or tunnel. This policy is applied from the service side, tunnel side, or both.

All these policies can be deployed centrally so that the entire network has similar function and behavior, or in some cases locally depending on a use case and how that particular site needs to be restructured. Policies based on traffic types like app-route, cflowd, control, data, and VPN membership can be configured and deployed.

Management Plane

Management of the SD-WAN architecture is provided by Catalyst Manager. This cloud-based controller manages all the WAN edge devices as well as Catalyst Controllers. Having a single pane of glass to effectively manage the entire SD-WAN architecture is crucial. Catalyst Manager uses the concept of templates. Similar to creating routing configuration for all the devices, feature templates define how each feature would look. Features consist of interfaces, routing protocols, security, ACLs, and so on. After these features are created, they are then grouped into something called device templates. These device templates are platform specific. Configuration consistency is key when managing modern fabric-based networks. Having all the templates based on site types and transport types makes the WAN more effective and predictable. Downtimes are more manageable and, in most cases, completely transparent. Catalyst Manager provides a scalable way to manage some of the largest SD-WANs of the world, consisting of thousands of sites spread across many continents.

Traffic Engineering

So far in this chapter, we have looked at modern fabric-based architectures and how underlay and overlay networks are built. Although these networks address many modern business challenges and are highly scalable, they also become highly complex. Consequently, having a controller to not only configure required devices with the right configuration and policies but also to effectively manage them is important. Designing an overlay may look easy, but as we go under the layers to underlay and to physical transport, the network may seem more complex. In most cases, there are high chances that traffic will trombone between the same two pairs of devices more than once. This does not create a loop in the network, because each time it traverses, it would have a different encapsulation. Designing the network in a way to avoid that situation as much as possible should be the goal of every architect. Looking at Figure 8-4, as we go deeper in the layers of multilayer architecture, we can see that the complexity increases and so does the overhead. Therefore, there is a big requirement of the higher MTU to be provided on the transport layer. The figure also shows constant encapsulation and decapsulation across different network points, and some may have more than one encapsulation, especially over the WAN.

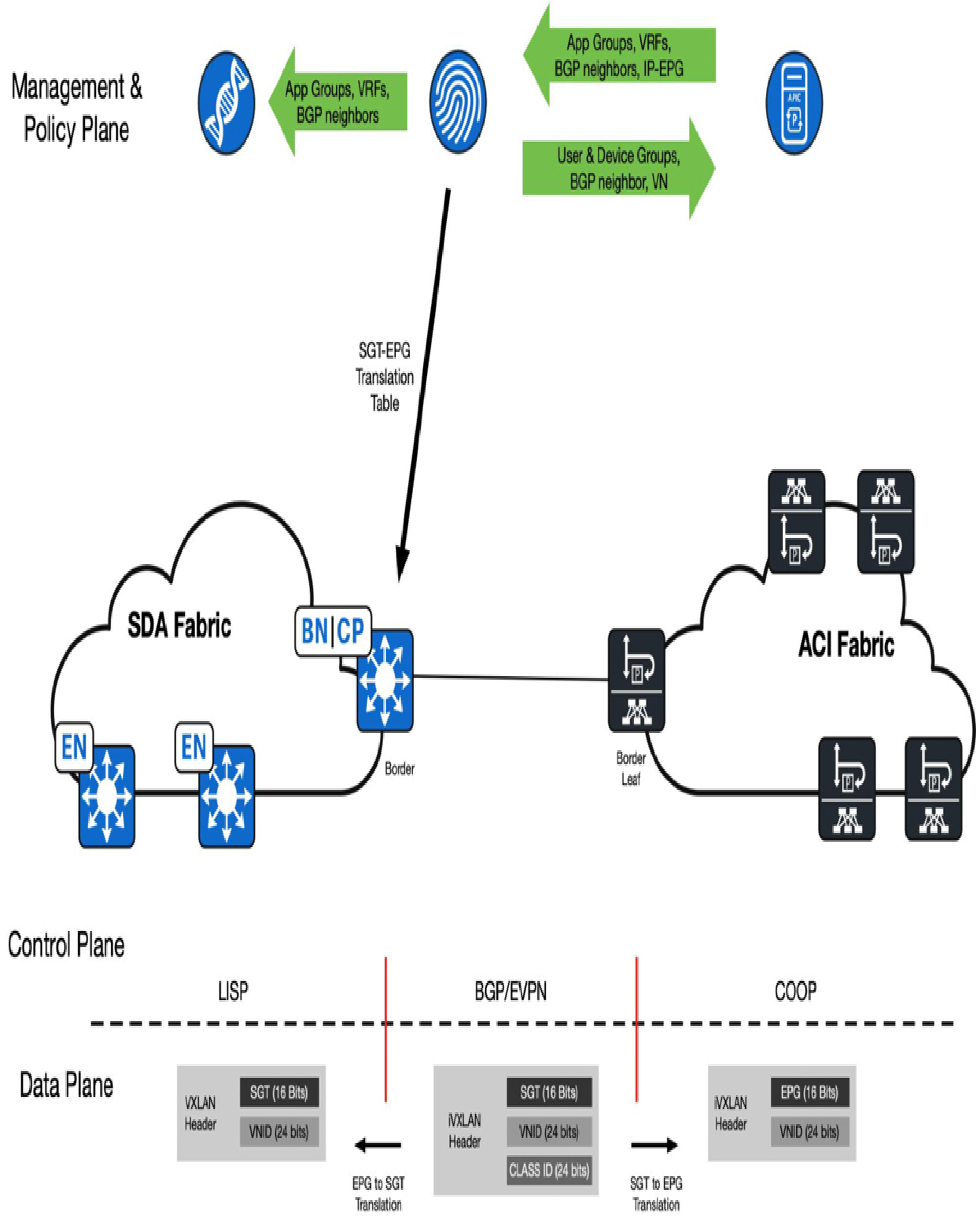


Figure 8-4 *Multidomain Multilayered Modern Network*

Traffic engineering in this environment is critical. Predicting when, where, and how the traffic will fail is required to ensure that business service-level agreements are met. Whether the failure is on the enterprise side or service provider side does not matter. When a network fails, the end user or an application is impacted. There may be financial transactions or some critical health and sensor data in the network that gets disrupted, causing undesired outcomes. We also need to take security into account when it comes to designing optimal traffic engineering. Securing networks is essential because one bad event can wreak havoc on the applications or endpoints or, even worse, isolate networks completely, taking entire businesses offline.

Next, let's look at some of the best practices on how to address these challenges in a most effective manner. We will look at components that help with most effective traffic engineering.

Network Design

A well-designed network can scale horizontally and vertically. Just like the network of the nervous system in the human body or an effective roadway hierarchy in a large city, designing hierarchical networks can lead to effective and scalable networks. Networks are 24×7 and need to maintain their standards of providing connectivity in any conditions. The higher the layer in the hierarchy, the more critical the network becomes. From a redundancy perspective, having up to two pairs of devices is sufficient, but as you go up one layer in the hierarchy, a block or modular network proves more effective than adding a tertiary device in the layer. Modular functional blocks are able to scale in both directions and are easy to replicate across multiple geographies. Effective network blocks and architectures can create effective choke points where a security layer can be inserted for further traffic inspection and enforcement. Figure 8-5 shows a multidomain functional network that is interconnected to provide a modular architecture that can be scaled up or down or multiplied based on the organization's requirements.

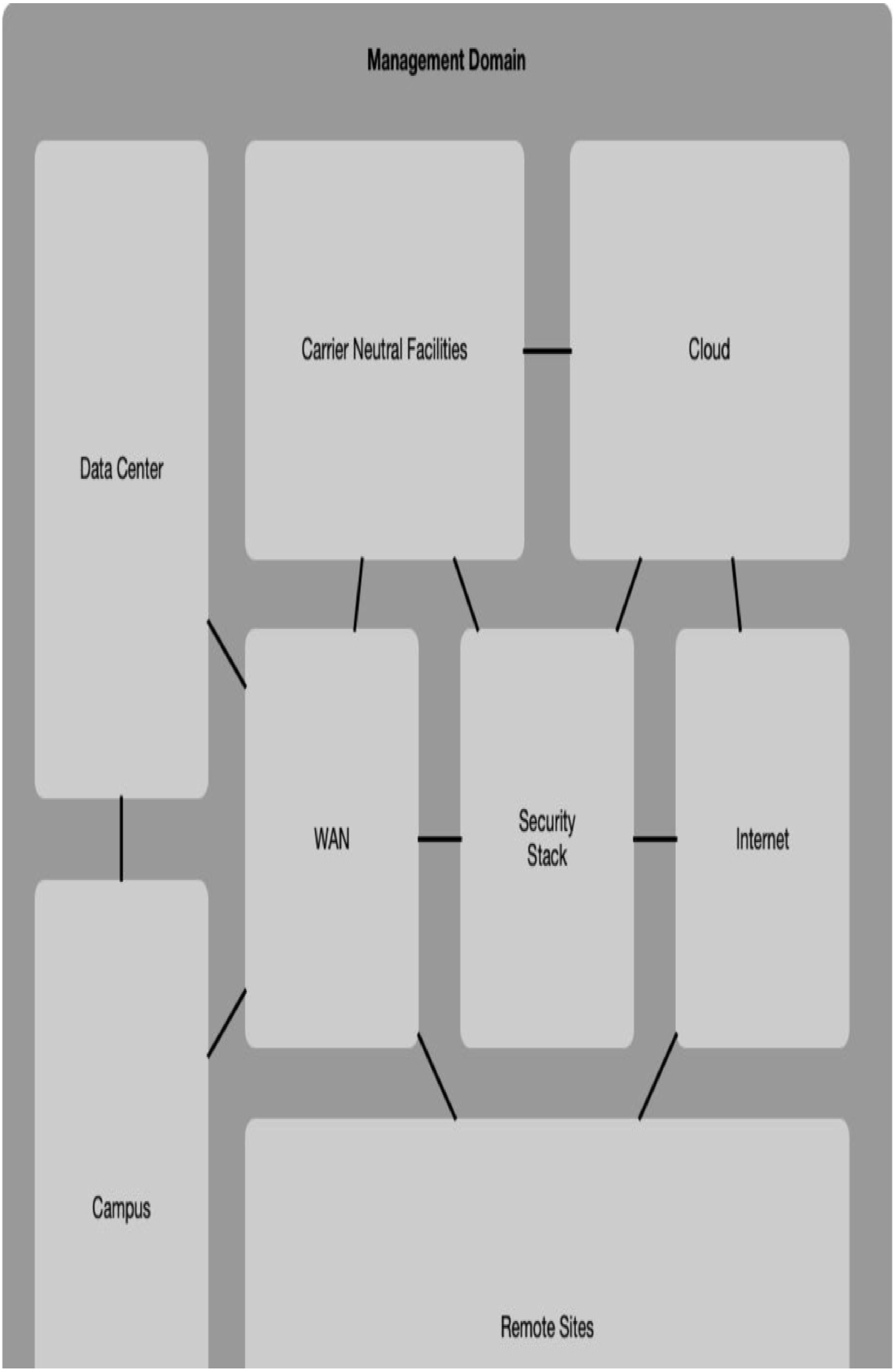

Management Domain
Data Center
Carrier Neutral Facilities
Cloud
WAN
Security
Stack
Internet
Campus
Remote Sites

Figure 8-5 *Multidomain Architecture*

Routing Protocols

Routing protocols play an important role in terms of traffic flow. Without them, there is no network. Earlier in the chapter, we discussed the importance of selecting the right routing protocols and what they provide in building a scalable underlay network. Having the right redistribution points and proper metrics to maintain the trueness of traffic path is critical. The adjacency of all routing protocols should be secured, and they should not be allowed to broadcast their protocol hello packets across unused links.

Traffic Flow Analysis

After the network is designed and the right routing protocols are set, it’s time to analyze the traffic. Traffic flow is important. Knowing where the users are accessing their data from and what route a particular flow is taking can provide great insight into capacity planning, security choke points, and any optimization strategies. If, at a given site, there is lot of cloud application access or Internet-bound traffic, it may be wise to leverage SD-WAN’s Direct Internet Access (DIA) feature to offload Internet-bound traffic locally from that site instead of backhauling to the data center. Similarly, if an organization has purchased dedicated cloud connections such as AWS DirectConnect, you need to make sure that all AWS-bound traffic does not take DIA and uses cloud connection instead. All of this can be determined from the flow analysis. For security purposes, the flow analysis can also give important clues when a network is about to be attacked. This happens when a traffic pattern suddenly changes and a huge spike appears between a source and destination that is not supposed to be there. The ability to catch this pattern and quickly act on it can save the organization a huge amount of trouble.

Traffic Management

Understanding the traffic flow leads to several other pieces. One of them is traffic management. Traffic management is not a big issue in enterprise networks, but it does impact service providers and cloud networks. For enterprises, traffic patterns are usually predictable. However, during certain events or parts of the year, traffic can spike upward for service and cloud providers. For example, during a major sporting event like the Olympics or Super Bowl, or a massive world event, live streaming and social media can spike traffic over service providers because everyone would like to watch and be updated about the situation. Similarly, during the holidays, when massive shopping deals are available, Internet traffic builds up on the cloud infrastructure as providers need to scale their infrastructure to accommodate the demands of the consumers. Looking at such historical patterns and proactively addressing those demands or having contingency plans will help with better traffic management and better end-user experience.

Load Balancing and Sharing

Often traffic management can be addressed by effectively load balancing and load sharing. Networks are designed with redundancy in mind. *Redundancy* means two of something, so if one part of a pair fails, the other one can take over. Historically, legacy Layer 2 and Layer 3 protocols were not effective in keeping networks active/active. With active/standby design, the primary device was almost always overloaded with the workload, and the secondary was just consuming power and was ready to take over the load in case the primary failed. Due to high reliability of the network devices, this failover was rare, and ROI of the investment was not realized fully. Today, that is not the case. With modern fabric-based architecture, all devices that load-share traffic effectively are active/active. Load sharing refers to unequal proportions of traffic sharing that is usually defined by flows. Load sharing occurs between network devices only and usually with an unequal cost path to upstream or downstream device.

Load balancing is trying to distribute equal load among different application servers. Usually, web servers or content servers have load balancers in front of them that are receiving all the requests from the end users or clients.

These load balancers will then take those requests and distribute them to the servers in round-robin fashion or to the server with a lower load factor. These load balancers keep track of workload on each server so that they can make effective decisions on where to offload new requests. In the holiday shopping scenario, when such peak times come, enterprises have automation and orchestration in place to spin up new servers on demand as load increases with user traffic, and they are added to the load balancers' pool automatically. As load decreases, load balancers start cleaning up and reallocating traffic to free up servers that can be shut down to conserve resources.

Quality of Service

Quality of service (QoS) has been in play for decades. QoS helps with traffic prioritization by ensuring business-critical and time-sensitive applications are allowed with priority in the network for better user experience. After the traffic analysis is done, QoS policies are placed where there is potential bandwidth congestion. QoS works on two parts: classification and enforcement. First, network traffic is classified by either IP subnet or by looking at the application. Once classified, that traffic is marked and tagged and propagated into the network. By default, all Layer 3 traffic markings are copied across to upper layer encapsulation protocols to maintain the classification. This classification is usually done as close to the source as possible. Once this marked traffic reaches a congestion or enforcement point, a policy map is configured that enforces these QoS markings based on the available bandwidth and prioritizes the traffic. This method has been used for decades, going back to a time when WAN bandwidth was very low and private circuits were very expensive. Enterprises did not want to waste expensive bandwidth for employees who were web surfing and watching videos. Today, however, QoS is mainly used as insurance on some critical links. One of the main reasons for this is high reliability and high bandwidth and lower cost of Internet circuits. No matter how much classification and marking are done, when the traffic hits the Internet, there is no guarantee of priority. The Internet is best effort service and follows a first in, first out (FIFO) model of traffic flow.

Bandwidth Planning, Congestion, and Oversubscription

Today’s networks do not have the same problems of high congestion, thanks to the cheaper and more accessible high-bandwidth Internet connections. With standardization of the Ethernet across all media and platforms, bandwidth has been increasing exponentially every few years. Whereas 100 Mbps connections were normal in the mid-2000s to 10 Gbps by 2015, in 2024, 400 Gbps connections were normal on most campus and data center switches. Bandwidth evolution has taken QoS out of the picture, and almost all of the switches and routers today support line rate throughput for normal IP traffic. Back in the day, switches were oversubscribed in terms of bandwidth, but with better ASICs and advancement in technologies, oversubscription and congestion are things of the past. QoS only kicks in once there is congestion. If there is no congestion, there is no need for the QoS.

No matter how much enterprise plans to future-proof their network infrastructure, there is an event in the world or industry that changes all of their planning. One such event in 2020 was the global pandemic, when the workforce started to work from home. All of a sudden, all organizations had to come up with a plan to continue their businesses remotely. Whole dynamics of traffic flow and pattern were shifted. There was a high demand in residential Internet traffic, and enterprise Internet and VPN firewalls were overloaded with thousands of employees trying to access their work resources from outside. This situation resulted in the adoption of cloud technologies at a faster rate that gave an ease to the bandwidth requirements. Increasing a 1 Gbps connection to a 10 Gbps connection is not easy because many things need to be changed—from the interface, optics, and in some cases, the fiber itself. However, shifting some of the important workload to the cloud helps in offloading external user traffic to the cloud without increasing local bandwidth or at least giving some breathing room for expansion.

Today, bandwidth planning is crucial. Historical and current traffic patterns and traffic flows are taken into account, and higher access interfaces with lower bandwidth caps are preferred as insurance. For example, most

Internet connections at a large site are 10 Gbps access with a 1 Gbps or 2 Gbps bandwidth cap. This ensures that businesses do not pay high costs up front, but in the event that they need more bandwidth, they simply have to ask their service provider to raise the cap.

Network Monitoring and Optimization

Once the traffic analysis is complete, and all management and planning are done, it’s time for monitoring and optimization. This is a fairly simple task but still an important one. Monitoring the network and application performance can lead to optimization in the network. As businesses grow, they will have a newer set of applications and requirements. Some may replace legacy on-premises applications with newer cloud- and SaaS-based applications. If an application consisted of about 15 percent enterprise-wide traffic, as the adoption of this SaaS-based application increases, so will the traffic shift. That 15 percent or more traffic will now start shifting toward the Internet or cloud connections. That shift needs to be taken into account, and optimization needs to be addressed. How the application fails over and the traffic shifts will need to be captured from network monitoring.

The task of monitoring and optimizing the network is an ongoing cycle and needs to be part of everyday process.

Policy and Security

Among the last components of traffic engineering are policy and security. In the earlier underlay and overlay sections, we discussed secure routing protocols and use of NAC, but at an overall network level, we need to understand how to secure entire networks. There are many physical and logical attributes related to network security. From a device perspective, the following aspects of the network need to be secured:

- **Control Plane Policing:** You need to prevent the device’s processor from being subjected to a distributed denial-of-service (DDoS) attack, making it stop forwarding data or slow down convergence.

- **Device Access:** You need to use the right amount of RADIUS or TACACS access with multifactor authentication (MFA) to ensure only authorized users are allowed to access a device.

- **Interface Protection:** We recommend adding an ACL to stop taking inbound connection requests, especially on public Internet-facing interfaces. This prevents inbound sniffing attacks and exploit vulnerabilities.
- **Time of Use Access:** After the devices are set up, they should not be accessed with full privilege access without a change control process. This is true for core or backbone switches because any misconfiguration can take down a large chunk of the network.
- **Security Audits and Firewall Rules:** A regular audit of firewall rules must be warranted to ensure there are no potential holes that can harm the network and overall system.

Global Internet

One of the last bits of the traffic engineering mechanism is the global Internet. Today, with IPv6 being adopted at a faster rate than before, almost all organizations are able to connect directly to Internet service providers (ISPs) and get full Internet routing tables. Direct access to Internet routing tables is good, but this access can also be dangerous if not planned properly. Internet peering is best if done with two or more ISPs. This approach provides protection for the organization’s public IP space; in this way, an outage on one ISP will not constitute an outage for the organization. The routing will take care and fail over to the secondary ISP. In planning such architecture, you must make sure that the proper route filtering and policies are in place. An organization does not want to inadvertently become a transit for Internet traffic. Policies need to be in place that advertise only the organization’s own prefixes and nothing else. The organization can choose to receive the entire or a partial Internet routing table and traffic-engineer prefixes of one ISP over the other. Planning Internet peering, although it may look simple, needs to be thought out carefully.

Geo-routing

Depending on compliance regulations, sometimes an organization may want to restrict access to its applications to different countries. For example, if a local bank does not have any branches outside of the state and does not offer any internal investment or banking products, it may not want to allow people from different countries to access its banking application. This is done by geo-routing. Since the Internet Assigned Numbers Authority (IANA) is responsible for allocation of IPv4 and IPv6 addresses to all organizations per region and country, the IANA maintains a comprehensive list of IP allocations to all the countries. Organizations can use this list to update their prefix list so that all inbound connections from restricted countries are denied and their networks can be further secured.

Today, geo-routing is also used for streaming content. Video content such as Netflix or YouTube uses geo-routing to publish local available content based on the location of the user. If a user is in India, suggestions would be provided based on that market; the process works similarly for users in the United States with their exclusive content. To avoid geo-routing issues, many users use third-party VPN services to tunnel their traffic through another country and pretend to be from a different region to leverage different content and/or access restricted applications. As networks and technologies advance, there will be mitigations and ways to detect such patterns.

Summary

In this chapter, you learned that routing and traffic engineering are fundamental for scalable, resilient, and secure networks. We looked into underlay routing protocols and their advantages and disadvantages. We also looked at various fabric-based overlay solutions and at traffic engineering and what components are required for traffic engineering to be useful.

References

1. IPv4 CIDR Report: https://www.cidr-report.org/as2.0

2. IPv6 CIDR Report: https://www.cidr-report.org/v6/as2.0

Chapter 9. Authentication and Authorization

In this chapter, you will learn about the following:

- What identity is
- Different types of authentication methods
- Enterprise authentication (dot1x)
- How to monitor authorization of endpoints

Overview

Identity verification is a concept that has been common in society for millennia, with one of the first-known references to an identity document being recorded as early as 450 BC. This reference can be found in the Old Testament's book of Nehemiah, recounting how Nehemiah, a cupbearer to Persian King Artaxerxes I, sought to contribute to rebuilding Jerusalem and asked the king for letters to the governors of the provinces situated west of the Euphrates River, guaranteeing him safe travel to Judah (Neh. 2:7–9). This narrative is an early testament to the use of official documents for secure passage, paralleling today's passports.

King Henry V can likely be credited in this domain in a modern sense, with the first passport used for travel in 1414. Since these times in history, the ability to prove one's identity has become a critical component of trustworthiness verification, with identity being required for deeds of land, verification of age, confirmation of trade certification or skill, and permission to access or be briefed on confidential information.

Fast-forwarding to today, identity has evolved significantly from its humble beginnings to the use of digital ledger technologies such as blockchain, where in banking, supply chain, and certain military applications, the technology allows for nonrepudiation.

Identity is a multifaceted concept that is defined differently across different academic fields, typically referring to the characteristics that identify and differentiate an individual or an entity:

- **Personal Identity:** An individual's self-conception, including traits, values, and beliefs, often discussed in philosophical and psychological terms.
- **Social Identity:** Group-based aspects of identity, such as cultural or ethnic affiliations, which are commonly explored in sociology and anthropology.
- **Digital Identity:** In information technology, the information used by computer systems to represent an external agent—for example, a set of data that uniquely describes a person, an endpoint, or a system's service, as well as the means of controlling access to certain resources within a system based on that unique data.

The inception of IT authentication dates back to the 1960s with the advent of password usage, coinciding with the emergence of the first computers. These initial computers were notably large, expensive, and inefficient by contemporary standards. Their ownership was confined to a handful of universities and large enterprises; however, demand was significantly high. In response, academic institutions like the Massachusetts Institute of Technology (MIT) pioneered time-sharing systems, notably the Compatible Time-Sharing System (CTSS), to facilitate concurrent resource utilization by multiple users on a single machine. Passwords were implemented to avoid everyone having access to everything on those initial systems.

In modern IT architectures, the term *triple A*—henceforth written as *AAA*, which stands for *authentication, authorization, and accounting*—is synonymous with the use and monitoring of verifiable identity, providing the right levels of privilege to access key resources. While this technology is not new, having first been proposed as an IETF draft in 1999, the right set of capabilities, use cases, and scenarios related to its deployability have evolved significantly over the years.

Within IP networks, concepts such as network access control (NAC) using AAA capabilities to limit access to the perimeter of a computer network and restrict lateral movement by applying access restrictions using identity represent a strong foundation in applying the methodologies of zero trust within organizations, as introduced in detail in Chapter 1, "What Is Zero Trust?"

A Broader View of Identity

Today the concept of identity expands beyond the simple use of username and password, which historically was the method of "securely" accessing information systems that were either local or connected to the Internet. Over time, it became apparent that the user-and-password pair was maybe not the most effective way to maintain security. It also became apparent that password sprawl can lead to scenarios where users would need to document or write down their credentials somewhere, potentially resulting in further scenarios where the credentials could be stolen, or a breach could take place as a result of the user's password having existed elsewhere—perhaps even in a public service in the Internet that had been compromised in the past.

Surprisingly, even today, identity management is often split into separate teams within the security entity of an organization, leading to challenges with the flow of information. This often occurs due to historical reasons, or due to policies within the company prior to the level of globalization that we see today. These approaches to identity management often lead to challenges in maintaining the right levels of operational rigor, accountability, and visibility to adequately handle incident response and align key security standards and strategy across the company's estate, or in conjunction with partners and third-party vendors.

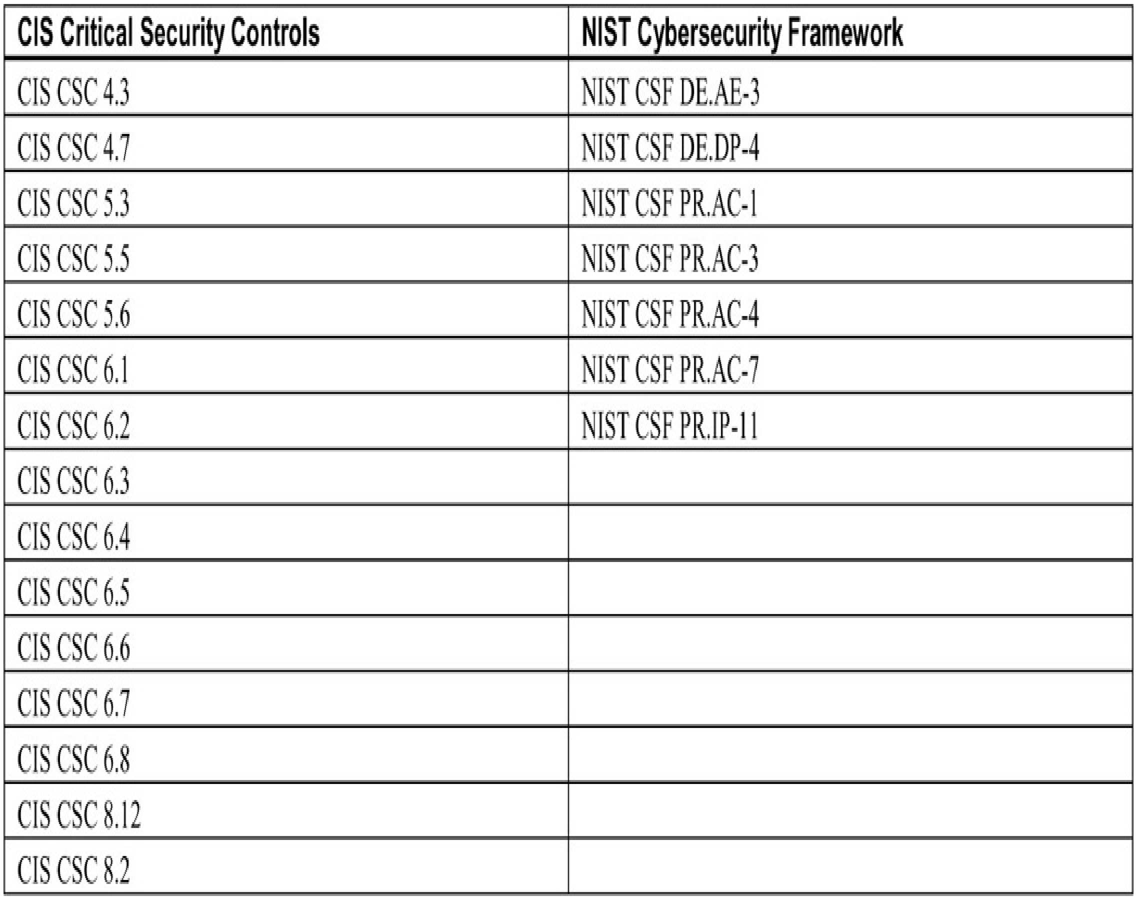
In today's architectures and systems, organizations can follow many standards and guidelines when it comes to identity and the correct levels of hygiene that should be applied from a security perspective.

In addition to systems such as workstations that are human operated (corporate endpoints), the number of Internet of Things (IoT) devices in enterprise networks is expected to exceed the number of corporate user endpoints in use by several counts. This change is being heavily fueled by the adoption of smart buildings, which represent additional challenges in the domains of identity, AAA, and profiling of these IoT devices. Based on our industry experience to date, these devices often lack the inherent security capabilities that are customary of corporate systems being used within organizations.

The National Institute of Standards and Technology (NIST) and the Center for Internet Security (CIS) have a rich set of best practices that should be considered in the context of identity while maintaining a zero trust–based architecture. These practices and recommendations transcend beyond simple authentication, including methods for logging and event retention and disabling orphaned or dormant accounts.

Table 9-1 provides an overview of security framework subcategories that are relevant to AAA. This framework is divided into five core functions: Identify, Protect, Detect, Respond, and Recover. Each function contains various categories with subcategories detailing individual outcomes.

Table 9-1 *Security Framework Subcategories Relevant to Identity and AAA*



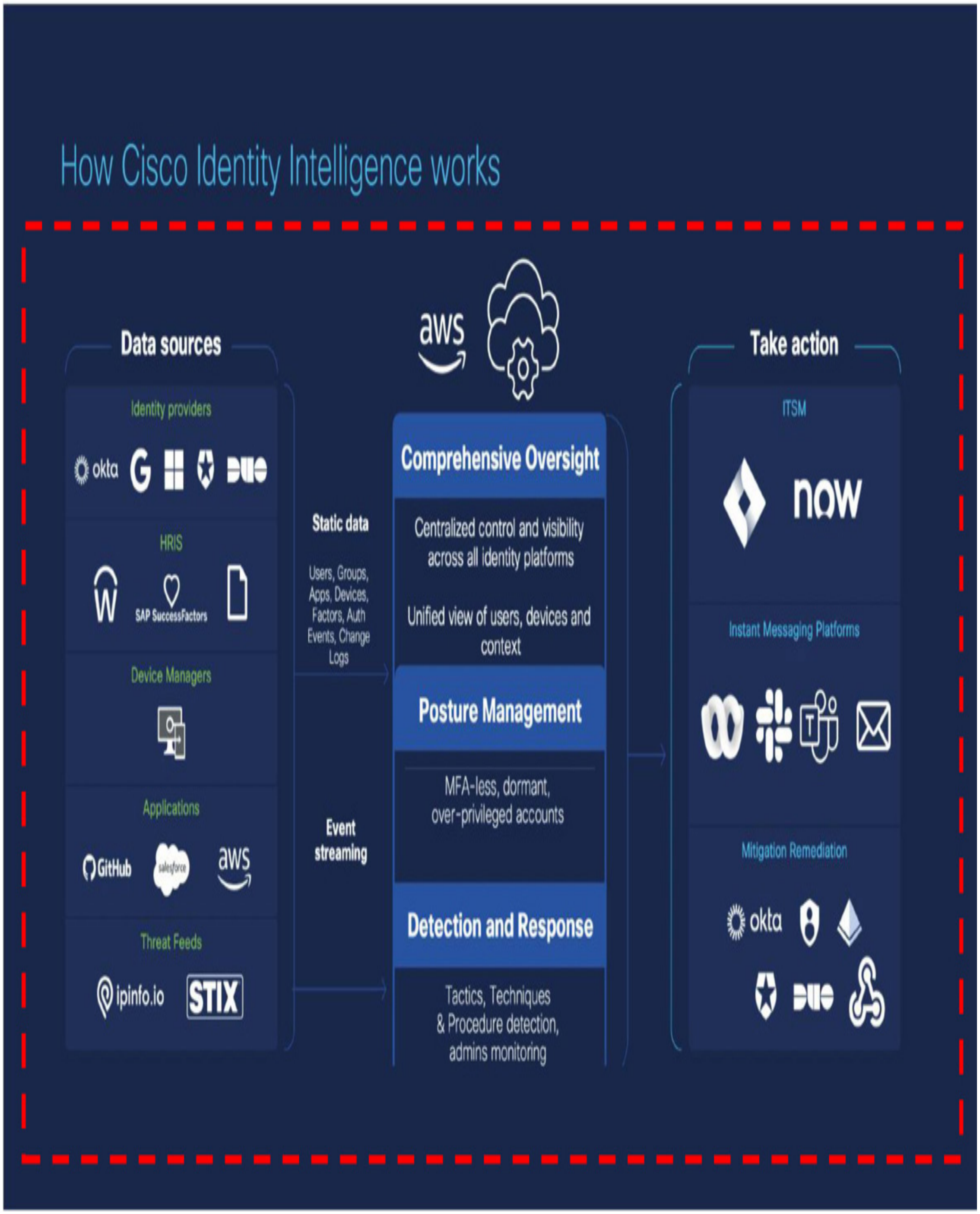
CIS Critical Security Controls	NIST Cybersecurity Framework
CIS CSC 4.3	NIST CSF DE.AE-3
CIS CSC 4.7	NIST CSF DE.DP-4
CIS CSC 5.3	NIST CSF PR.AC-1
CIS CSC 5.5	NIST CSF PR.AC-3
CIS CSC 5.6	NIST CSF PR.AC-4
CIS CSC 6.1	NIST CSF PR.AC-7
CIS CSC 6.2	NIST CSF PR.IP-11
CIS CSC 6.3	
CIS CSC 6.4	
CIS CSC 6.5	
CIS CSC 6.6	
CIS CSC 6.7	
CIS CSC 6.8	
CIS CSC 8.12	
CIS CSC 8.2	

DE.AE-3 falls under

- Function: Detect (DE)
- Category: Anomalies and Events (AE)
- Subcategory: DE.AE-3

Subcategory DE.AE-3 states "Event detection information is communicated to appropriate parties."

Current modern innovations around identity, including a consolidated view and perspective across the identity attack surface within a business, provide a bird’s-eye view of the potential threat landscape that exists across the disparate platforms that a user or machine account may be interfacing with. Figure 9-1 shows some the identity sources that Cisco Identity Intelligence uses to gain such perspectives.



How Cisco Identity Intelligence works

Data sources

Identity providers

HRIS

Device Managers

Applications

Threat Feeds

Static data

Users, Groups, Apps, Devices, Factors, Auth Events, Change Logs

Event streaming

Comprehensive Oversight

Centralized control and visibility across all identity platforms

Unified view of users, devices and context

Posture Management

MFA-less, dormant, over-privileged accounts

Detection and Response

Tactics, Techniques & Procedure detection, admins monitoring

Take action

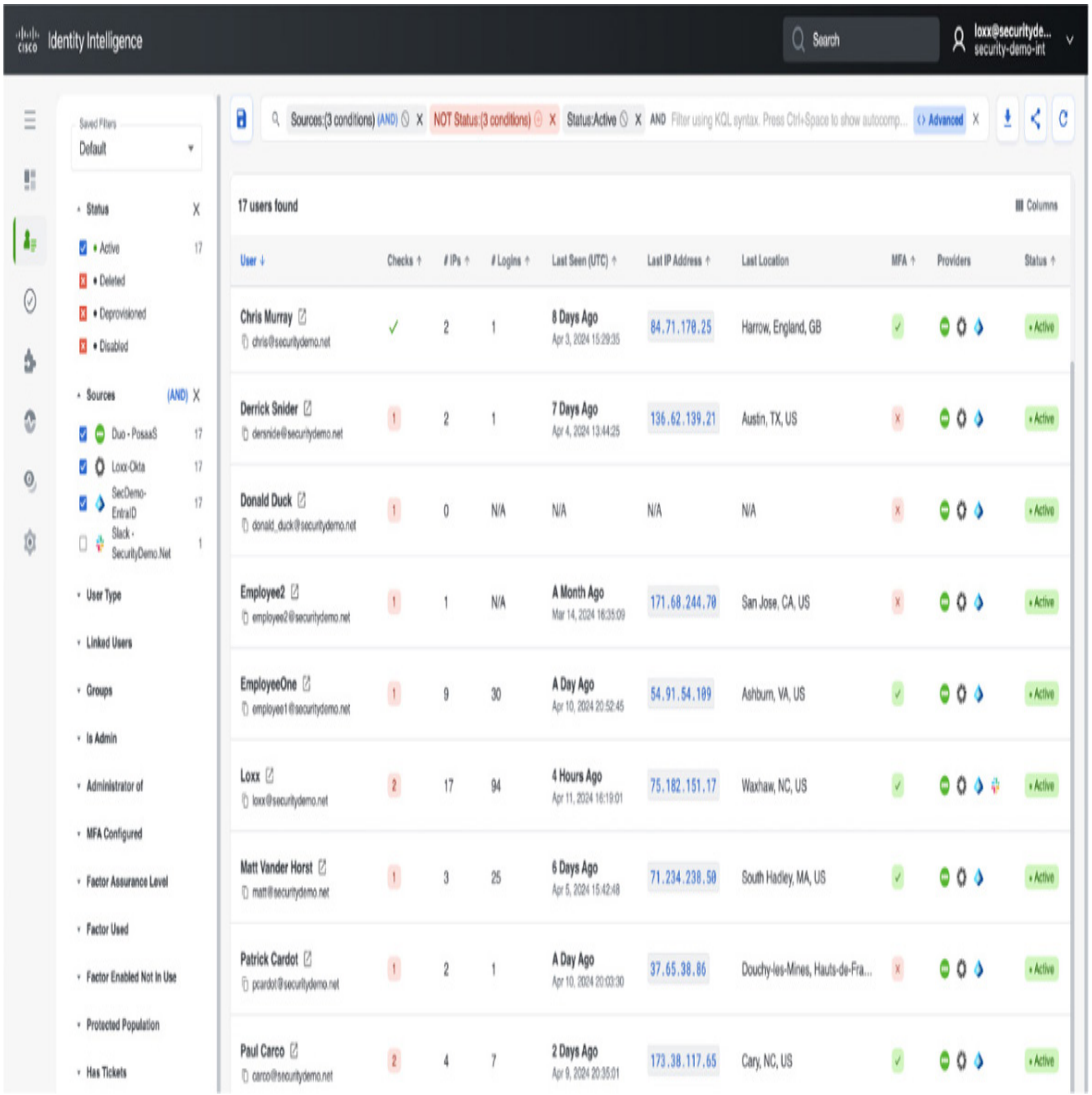
ITSM

Instant Messaging Platforms

Mitigation Remediation

Figure 9-1 *Cisco Identity Intelligence (formerly Oort) Utilization of Multiple Data Sources*

Cisco Identity Intelligence (Oort) provides a means to achieve identity threat detection and response capabilities by linking with the different identity providers and visualizing an overview of the consolidated logins per user, as can be seen in the dashboard shown in Figure 9-2, and potential anomaly-based behavior, such as physical location mismatches with logins.



☑	Active	17
☒	Deleted	
☒	Deprovisioned	
☒	Disabled	

Sources (AND)

☑	Duo - PosaaS	17
☑	Loxx-Okta	17
☑	SecDemo-EntraID	17
□	Slack - SecurityDemo.Net	1

Sources:(3 conditions) (AND) X NOT Status:(3 conditions) X Status:Active X AND Filter using KQL syntax. Press Ctrl+Space to show autocomp...

17 users found

User ↓	Checks ↑	# IPs ↑	# Logins ↑	Last Seen (UTC) ↑	Last IP Address ↑	Last Location	MFA ↑	Providers	Status ↑
Chris Murray chris@securitydemo.net	✓	2	1	8 Days Ago Apr 3, 2024 15:29:35	84.71.170.25	Harrow, England, GB	✓		Active
Derrick Snider dersnide@securitydemo.net	1	2	1	7 Days Ago Apr 4, 2024 13:44:25	136.62.139.21	Austin, TX, US	✗		Active
Donald Duck donald_duck@securitydemo.net	1	0	N/A	N/A	N/A	N/A	✗		Active
Employee2 employee2@securitydemo.net	1	1	N/A	A Month Ago Mar 14, 2024 16:35:09	171.68.244.70	San Jose, CA, US	✗		Active
EmployeeOne employee1@securitydemo.net	1	9	30	A Day Ago Apr 10, 2024 20:52:45	54.91.54.109	Ashburn, VA, US	✓		Active
Loxx loxx@securitydemo.net	2	17	94	4 Hours Ago Apr 11, 2024 16:19:01	75.182.151.17	Waxhaw, NC, US	✓		Active
Matt Vander Horst matt@securitydemo.net	1	3	25	6 Days Ago Apr 5, 2024 15:42:48	71.234.238.50	South Hadley, MA, US	✓		Active
Patrick Cardot pcardot@securitydemo.net	1	2	1	A Day Ago Apr 10, 2024 20:03:30	37.65.38.86	Douchy-les-Mines, Hauts-de-Fra...	✗		Active
Paul Carco carco@securitydemo.net	2	4	7	2 Days Ago Apr 9, 2024 20:35:01	173.38.117.65	Cary, NC, US	✓		Active

Figure 9-2 *Cisco Identity Intelligence (Oort) User Dashboard*

Authentication and Authentication Methods

Let's revisit the core concepts of credentials and authentication, examine diverse authentication methods, and look at real-world examples to anchor this discussion in practical scenarios. *Authentication* is defined in NIST Special Publication 800-207; as part of this definition, there is a mandate to ensure that all access requests must be authenticated, without any banner exceptions that could result in implicit trust being granted. NIST further defines authentication as a key process responsible for the identification of users and systems, which is a cornerstone in achieving a zero trust architecture.

When a user is attempting to log in to a website from a Software as a Service (SaaS) provider, connecting a virtual private network (VPN), logging into a desktop environment, or connecting to a network device, the most simplistic way to think of authentication is as the binary permission or rejection to a resource based on information provided by the user, such as a username and password. Fortunately, such methods are not commonplace anymore. Historically, the use of Password Authentication Protocol (PAP) performed just that, executing a two-way handshake, sending both username and password credentials in cleartext (see Figure 9-3).

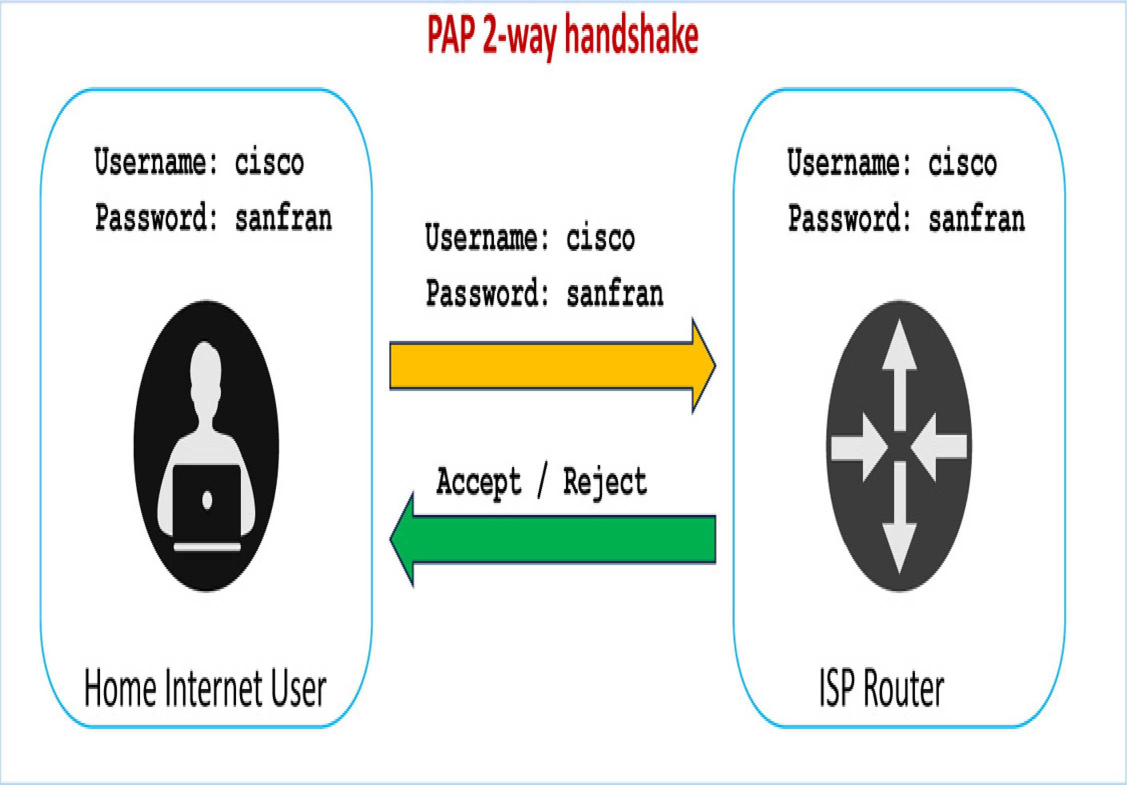


Figure 9-3 *PAP Authentication*

PAP is one of the most basic of methods of authentication and for many functions today remains a relic of history. Thankfully, today a broad array of other authentication options is available to offer users a reliable and more robust means of authenticated access to applications and networks.

In the following sections, we’ll take a closer look at how this activity is achieved in different contexts and with different systems to provide a rounded view of how authentication looks today in modern information technology landscapes.

Local Authentication

Prior to the existence of distributed networks, stand-alone systems required security mechanisms to ensure that a viable audit trail was available to indicate which users were performing which data entry functions to provide accountability. To provide that level of differentiation on information systems, the concept of user credentials was created. This means that for each individual system, unique usernames and passwords or other forms of authentication are configured and stored directly on the device itself. These types of credentials are referred to as *local credentials*. They’re the core element of local authentication, a process where the authentication of user credentials is handled directly by the local system or device without the need for a centralized server.

While this method worked relatively well on stand-alone systems in early days of computing, security requirements and practices associated with the deletion of users who left the company, new individuals joining the company or department, and challenges of password change management made this approach relatively cumbersome to deploy at scale.

In today’s systems, local credentials often still exist but are, in many cases, deployed as an initial starting point to facilitate access for day-0 installation of systems, or as an emergency fallback login method in scenarios where more elaborate central authentication systems may be unavailable. Example 9-1 shows the output of a Cisco IOS XE switch configuration of a local username and password. Here, the username can be read in cleartext as *sdaadmin*, followed by a definition of the encryption type with the command syntax *secret 9*, which refers to the scrypt hashing algorithm that offers enhanced security against brute-force attacks, and finally the encrypted password hash represented by the 59-character long string.

Example 9-1 *Local Credentials Configured on a Switch*

```
Branch-FIAB#show run | section username
username sdaadmin secret 9 $9$dfyZHtaK9g1BXU$SDDJFKLDd88NuwYuSMlK
Branch-FIAB#
```

Note that even when centralized server–based account and authentication management methods such as those enabled by LDAP, RADIUS, Diameter, and others are used in conjunction with systems that have local credentials configured, this doesn't negate the insecurity of the local credentials' presence. Most employers must deal with attrition and change of staff over time, and they often do not apply the measures necessary to ensure that the local admin accounts that exist on common systems are updated to leverage new usernames and passwords when individuals depart or do not update on a regular schedule.

Besides the considerable disadvantages, local credentials have also some advantages. For example, they limit the potential impact of not being able to access a system due to compromised credentials. They are also critical for securing access to devices that may not always be connected to the network.

Centralized Server–Based Authentication

As mentioned, maintaining local authentication data and credentials may have some advantages for privately managed systems or devices that do not need to connect to a broader computer and infrastructure network. However, once a larger environment of connectivity is needed, taking such an approach has its limitations.

The introduction of centralized server–based authentication methods evolved from the popularity of dumb terminals, which were used to connect to a mainframe system, whereby each user would log in with their own username and password. Now, this approach, while using a central system, may not necessarily be considered the same as server-based authentication today. Those terminals were essentially communication nodes that utilized RS-232 serial connections instead of Ethernet to access the central system —likely more comparable to the connection that you would use when connecting to the console of a router or switch. However, leveraging a remote terminal or system to access a central resource became an expectation for users, which resulted in a progression in technological innovation.

Over time, the evolution of authentication technologies has led to the implementation of servers leveraging protocols like Kerberos, LDAP, Diameter, and RADIUS to enhance security and streamline user access management. Diameter is predominantly used in the mobility sector for 4G and 5G networks, whereas RADIUS remains the prevalent choice for enterprise network authentication.

RADIUS, or Remote Authentication Dial-In User Service, is a protocol designed to manage three critical network security functions: authentication (verifying if a user can access the network), authorization (determining the user's privileges on the network), and accounting (logging the user's network activity). These functions are commonly referred to as AAA, as noted previously.

The protocol's main advantage lies in its ability to centralize AAA functions, making it easier to manage access across various networking infrastructures and locations. RADIUS is also an open-standard protocol, ensuring wide compatibility and adoption. RADIUS uses UDP ports 1812 for authentication and 1813 for accounting; its fundamentals are outlined in the IETF standard RFC 2138, established in 1997.

RADIUS uses a client/server model, and its three primary components are

- **Client/Supplicant:** This is the device or user seeking access to a network. More precisely, it is software built in or installed ad hoc on an endpoint's operating system that passes information about a user (username, password, or certificate) to a second component, namely the network access server.

- **Authenticator/Network Access Server (NAS):** This is basically the gateway or network access device between a user and the network to which the user is seeking access. In the RADIUS client/server architecture, the NAS acts as the client.

- **Authentication Server:** The authentication server ensures that the user is allowed to access the network and that the user does so with the proper permission levels. In the simplistic scenario depicted in Figure 9-4, the RADIUS server performs the authentication server function. In more advanced architectures, multiple RADIUS servers may be in use with dedicated functionality defined per grouping of users (Corporate/IoT/Guest).

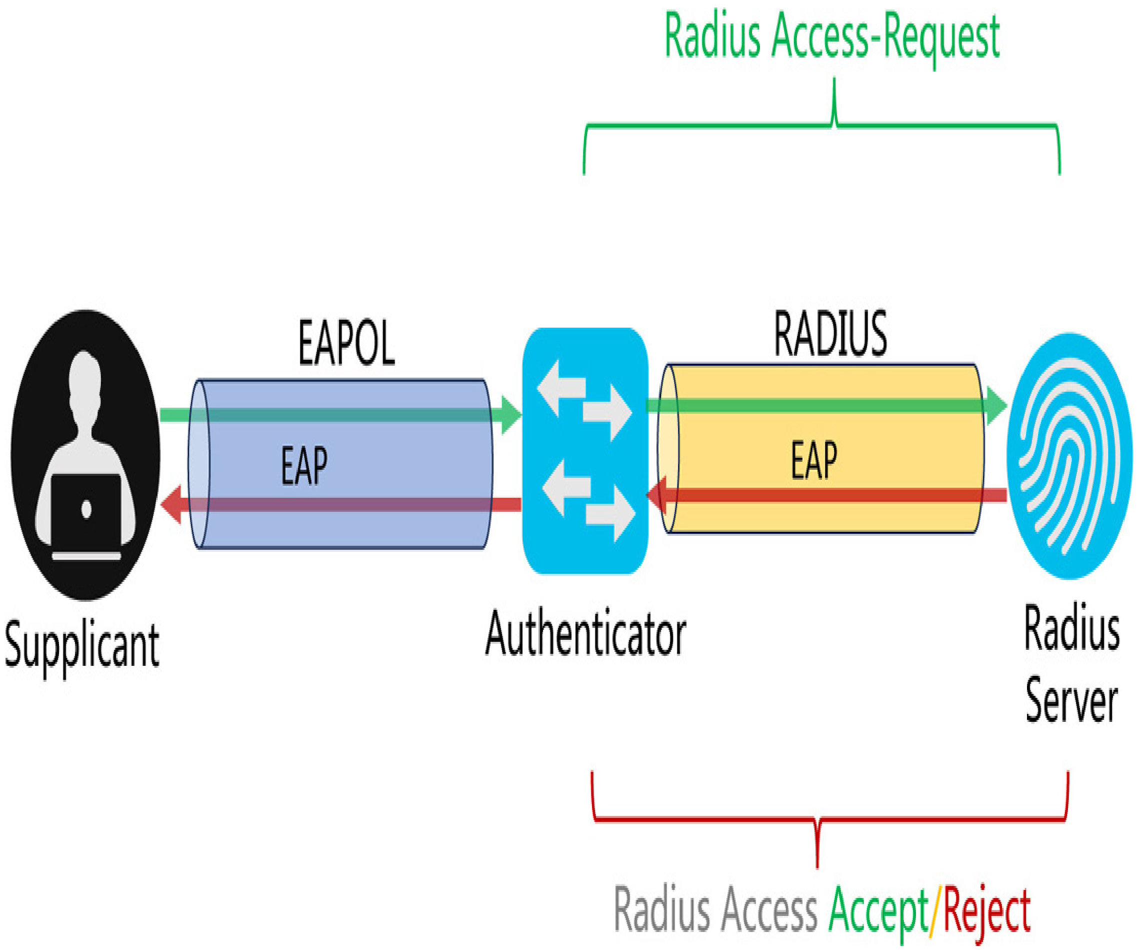


Figure 9-4 *Server-Based Authentication*

RADIUS and its components facilitate the integration of core user identities stored in a directory such as Microsoft Active Directory (AD), Microsoft Entra ID, OpenLDAP, or the RADIUS server's own database with the network infrastructure.

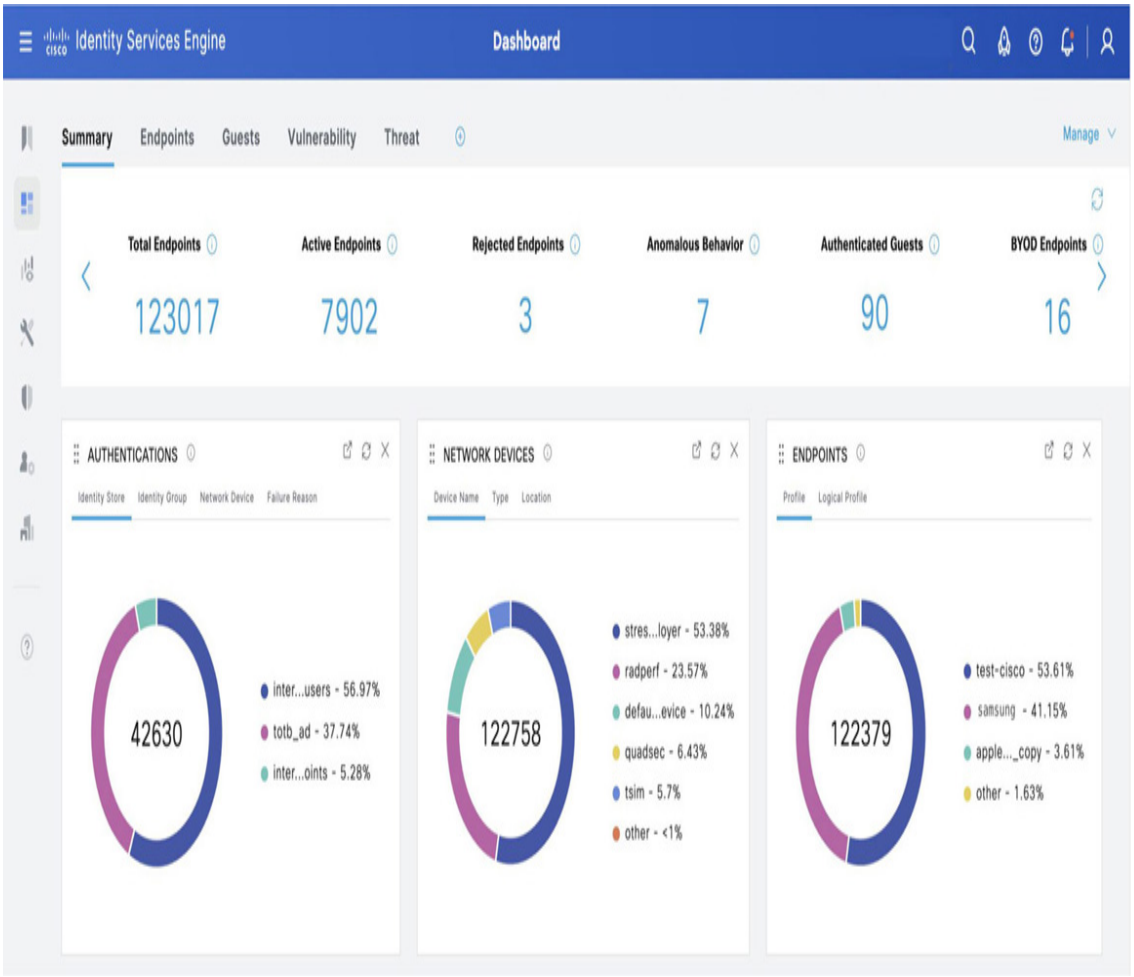
In enterprise network environments, using LDAP or Microsoft Active Directory solutions is very common. Cisco Identity Services Engine (ISE), which is referenced frequently in this chapter, is an example of an enterprise-grade RADIUS server solution. Advanced integration options are available for users within the Cisco ISE RADIUS server in both on-premises and cloud-based options of deployment.

When an organization is planning to deploy a centralized AAA server architecture, it needs to carefully plan the placement of RADIUS servers and directory services servers. RADIUS, as a protocol, is relatively resilient, able to withstand multiple seconds of delay under some circumstances. For this reason, the placement of RADIUS and directory services needs to be measured in terms of worst-case cumulative round-trip time between client and backend server to ensure that the planned deployment is feasible and will not result in a disruption of service or other issues for end-user devices.

To ensure an efficient authentication process, it is advisable to aim for a completion time of under 5 seconds. This recommendation is made considering that the round-trip time (RTT) between the network access device (NAD) and the AAA server (for example, Identity Services Engine) is only one element of the total time to account for. It is therefore crucial to also consider the time consumed by subsequent operations, such as those RADIUS requests sent to the AAA server that might entail an external identity and credentials lookup to LDAP or Active Directory. The configuration settings of the NADs are a determining factor in this equation. Consider, for example, the default RADIUS timeout of 5 seconds on a Cisco Catalyst 3850 switch. If ISE’s response time exceeds this duration, the switch is programmed to reissue the request. Therefore, maintaining a total response time under 5 seconds is key to preventing such retransmissions and ensuring a streamlined authentication process.

Understanding key thresholds and benchmarks for your network deployment is important. Without having a solid understanding of what “good” looks like, it is almost impossible to understand whether a software-defined network architecture that utilizes authentication infrastructure such as RADIUS has properly recovered from an outage scenario or if there are still lingering issues.

For example, Cisco ISE provides numerous dashlets to gain a view and perspective of the current overall health of a deployment. Although these figures look useful and helpful—which they are in small-scale deployments—when you're looking at larger deployments with hundreds of thousands of users, it is relatively easy to miss an outage scenario by just looking at the main page, as shown in Figure 9-5.



Total Endpoints 123017

Active Endpoints 7902

Rejected Endpoints 3

Anomalous Behavior 7

Authenticated Guests 90

BYOD Endpoints 16

Figure 9-5 *ISE Summary Dashboard*

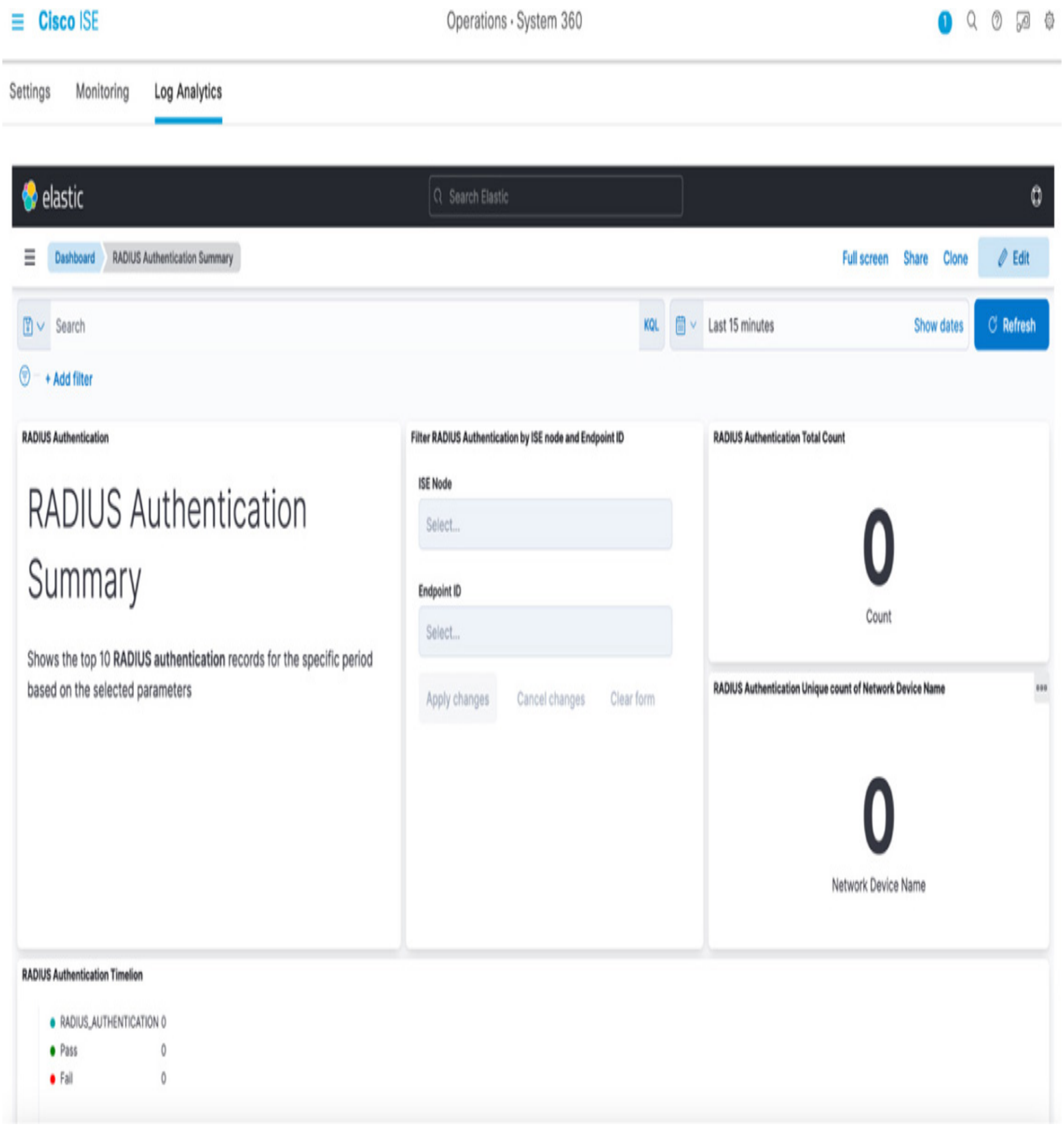
To provide operators with a more in-depth and powerful view of authentication and activities that are happening on the network, Cisco Identity Services Engine also provides the capability to build out custom metrics based on logs and events and queries that are configured.

ISE System 360, introduced in version 3.2, enhances the ISE with advanced monitoring and log analytics capabilities:

- The Monitoring feature allows for comprehensive oversight of various application and system metrics, including the key performance indicators (KPIs) for all nodes within a deployment, all accessible from a centralized dashboard.

- Log Analytics offers a versatile analytics platform designed for detailed examination of endpoint authentication, authorization, and accounting processes, in addition to profiling syslog data. This feature facilitates the analysis of health summaries and process statuses for Cisco ISE, enabling administrators to derive actionable insights from system behavior and performance.

Log Analytics is depicted in Figure 9-6, and as mentioned, it can offer targeted data sets for analysis and assessment to get the right view of what may be happening within the network authentication architecture.



Shows the top 10 **RADIUS authentication** records for the specific period based on the selected parameters

Figure 9-6 *Log Analytics View in ISE*

Service Accounts

One example of credentials that are often challenging to maintain and administer in an organization is service accounts, much like emergency local credentials that may exist on a system for use in a scenario where server-based authentication is not possible. A requirement of service accounts is to be used persistently for some form of ongoing operation, usually a machine-to-machine authentication, or execution of a script or program that may be using an API as part of its workflow.

One such scenario in the networking world could be considered the authenticated access point workflow, where the authentication methods EAP-FAST or PEAP may be used. With these two protocols, username and password are used as part of the authentication flow. In common enterprise architectures, this username and password pair would not be present on the switch on which the access point is performing its authentication, but rather upstream in a third-party information store such as Microsoft Active Directory.

Maintaining these user accounts and handling the correct change management and credential updates for these service user accounts on a regular schedule are often key security requirements for adhering to corporate security policy.

Because these accounts tend to have access to key and critical infrastructure components, the group of individuals who have access to these credentials at any time should be limited to a "need-to-know" basis.

When Using Service Accounts Goes Wrong

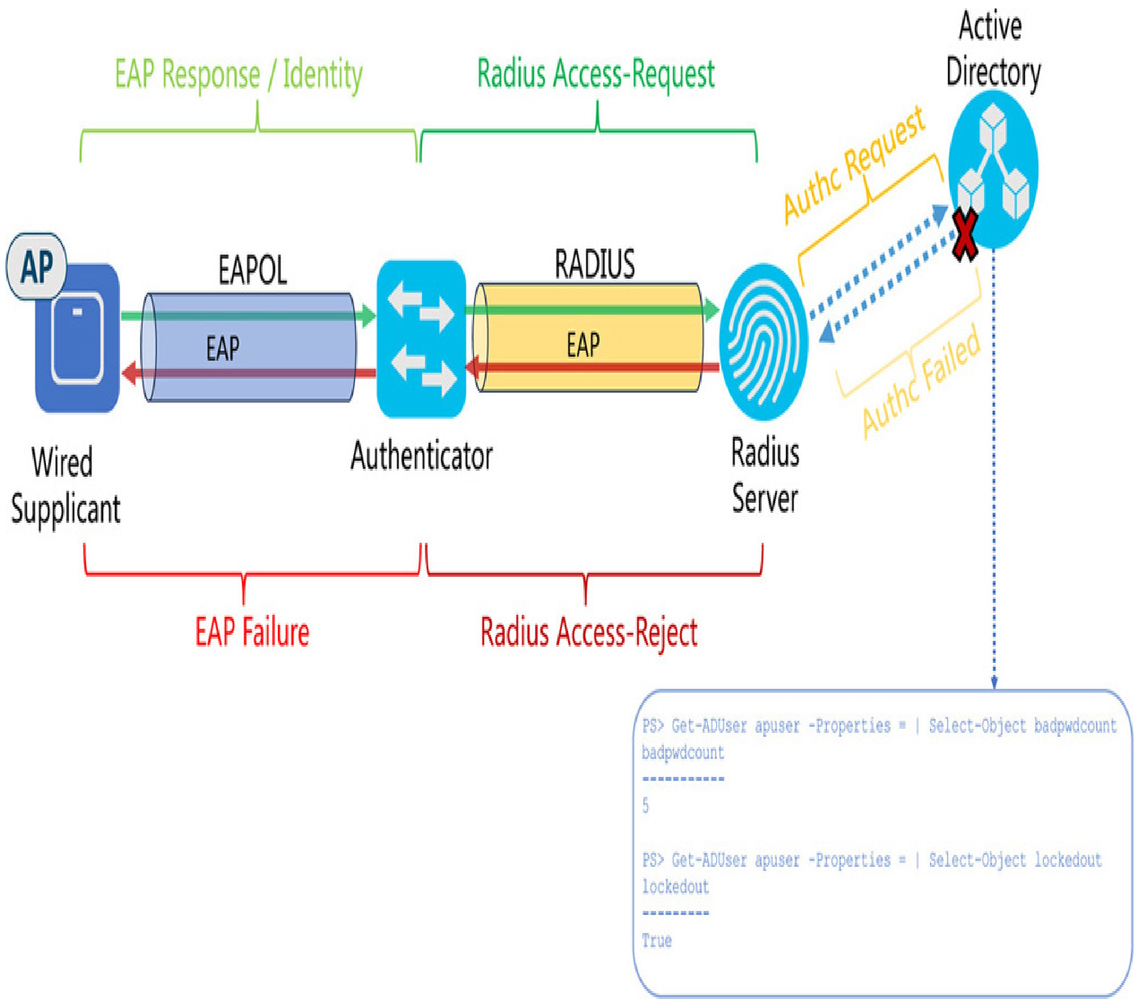
Some years back Cisco Professional Services team members were engaged to support a defense industry customer who was deploying access points throughout one of its secure sites. Because this was a secure network, all switchports were enabled with IEEE standard 802.1x framework, which defines the use of EAP encapsulation to allow a client device (supplicant) to perform secure authentication. This specific scenario required closed authentication, permitting only devices to onboard that used certificates (via EAP-TLS) or username and password for some scenarios for service users. At this time, Cisco access points were only able to support EAP-FAST as an authentication method. For the first month of operations, the access points were authenticating correctly against their connecting switch ports, and connecting clients were happily connecting to the Wi-Fi network and working in production.

One afternoon, the network help-desk phones started to light up with complaints about Wi-Fi coverage. The callers complained that their connectivity was poor. Even after they moved to another area of the building, their connection would work for a while and continue to worsen.

Looking into their monitoring systems, the network operators at the help desk began to see what was going on: access points that were previously online and registered by the wireless LAN controller were disconnecting. Of the over one thousand access points installed on the site, every time that the operators refreshed their view of connected access points, the number of connected access points became smaller and smaller. This scenario explained what was being reported by the users. As their access points were going offline, they needed to rely on residual coverage from the next-best access point on the floor, but with the devices rapidly going offline, soon all users' connectivity would be gone.

Taking the next step, the network operations team decided to look at a switch port that a formerly working access point was connected to. To their surprise, the team observed that the port was actually in a state of Authc (Authentication) Failed, as shown in Figure 9-7. The operators found this situation to be confusing because the credentials were configured from the central wireless LAN controller, and the team had not seen any admin user changes registered from that system over the past weeks.

Looking more closely at both the RADIUS server logs and the Active Directory status for the users, the team quickly identified what was amiss: the service user was locked out of the Active Directory server as a result of having set the wrong password too many times. The AD server administrator quickly unlocked the user, which resulted in the access points coming back online again. However, this did not explain how the situation happened in the first place.



```
PS> Get-ADUser apuser -Properties * | Select-Object badpwdcount
badpwdcount
-----------
5

PS> Get-ADUser apuser -Properties * | Select-Object lockedout
lockedout
---------
True
```

Figure 9-7 *Locked Service Account Used for the Access Point’s Wired Dot1x*

By tracing through the RADIUS logs from the past 24 hours, the team quickly identified something unusual; they identified a wired user with an Apple laptop attempting to log in to the network with the access point service user’s credentials. From the logs, the team could identify that the login attempt took place from a user in the IT department on the fourth floor, on a switchport that was cabled to a specific user’s office. After the team reached out to that individual, it became apparent that the user was performing some connectivity tests with a new laptop and figured that it would be quicker to use the respective service user credentials that he thought to have remembered properly rather than waiting for a new test user account to be created for his activities. Unfortunately, the fact that the user was attempting to log in to the service account with the wrong password but still with the correct username led to the calamitous events that took down the access points in the building.

Because the impact of this outage resulted in a significant service disruption for the business, a postmortem analysis took place, resulting in the following conclusions:

- Service accounts should be certificate-based wherever possible.
- Usernames and passwords should be cycled every three months.
- Certificates for service accounts should be cycled every six months.
- RADIUS policy rules should be created to quickly match exceptions to normal service account usage (connection method and so on).
- Service accounts must never be actively used for testing functions outside of their primary use.

x.509 Certificate-Based Authentication

Using the x.509 standard for public key infrastructure—certificates instead of username and password—has become the preferred means of machine-to-machine communications; this approach establishes trusted authentication for many use cases in the IT industry today. The x.509 certificates rely on asymmetric cryptography (public key cryptography using public-private key pairs), where *asymmetry* refers to a clear separation and distinct functions of different cryptographic keys and roles in the authentication process—for example, distinct cryptographic keys (a private and a public one). The entity that verifies the identity uses the public key and does not have access to the private key. A certificate authority (CA) issues digital certificates that associate public keys with identities, while users and systems use these certificates to prove their identities.

In systems relying on username/password type credentials, the authenticator has access to the password itself, at some point, or an equivalent set of data (such as a hash) that can be used for verification purposes. This presents an inherent security risk because the authenticator possesses the credentials required to potentially compromise the user's identity.

Conversely, user certificates employ a different approach. These certificates are conferred by a trusted certificate authority that validates and affirms the association between an individual's verifiable physical identity and a unique cryptographic public key. The role of the verifier in this framework is notably separate; it is enabled to authenticate the identity of the user by referencing the certified linkage of the public key without acquiring the capabilities to forge the user's identity.

To summarize the essence of user certificates, their primary function is to bifurcate the responsibilities between entities that establish a user's digital identity—that is, the process of translating an individual's physical identity into a digital counterpart—and those entities that are responsible for user authentication.

This architectural separation not only enhances security but also paves the way for the implementation of digital signatures, which provide an additional benefit of nonrepudiation, ensuring that a user cannot credibly deny the authenticity of their digital transactions or communications. Within cloud environments, the use of certificates to access tenants and hosts has become ubiquitous and reduces the ability for a malicious user to breach publicly hosted resources through brute-force dictionary attacks. A certificate can generally be validated as secure if it come from a trusted certificate authority, with a limited lifetime or duration of validity. The more sensitive the resource that is receiving the certificate, the more frequent the certificate renewal requirement may be.

To further secure the use of certificates, authenticating systems can use a certificate revocation list (CRL), which is a list of certificates that have been revoked prior to their expiration. These systems then can confirm if a particular certificate has been revoked or removed, thereby failing authentication of such "deny listed" devices. While using CRLs may sound like a great idea, it is not without its challenges. In a small deployment, there may be only a handful of devices that have had their certificates' pre-expiry period revoked. In a large deployment, the number of revoked certificates may increase quickly, leading to a very large revocation list that needs to be downloaded, as shown in Figure 9-8.

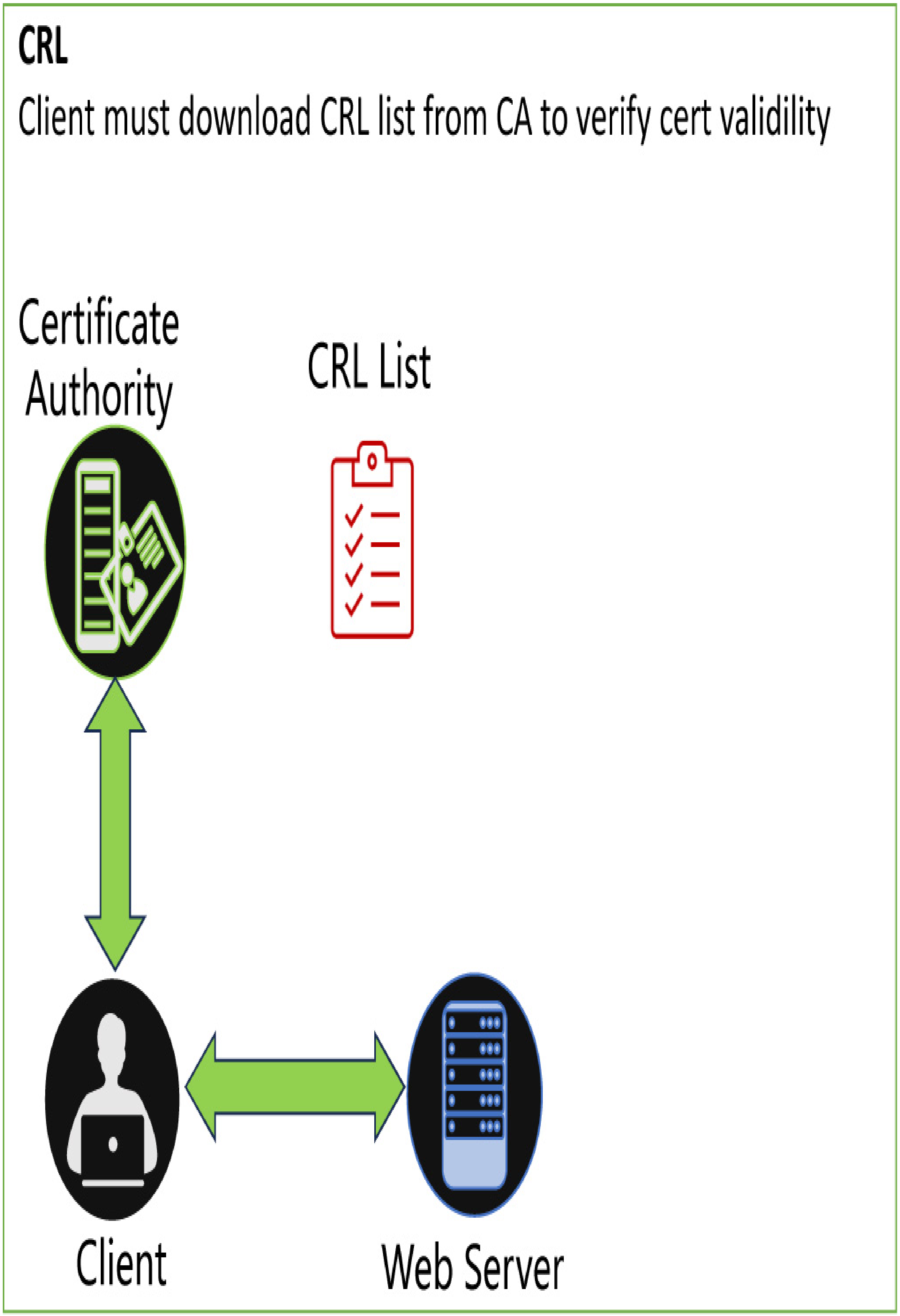

CRL
Client must download CRL list from CA to verify cert validility
Certificate Authority
CRL List
Client
Web Server

Figure 9-8 *CRL-Based Certificate Validation*

The alternative to the use of CRLs is to use the Online Certificate Status Protocol (OCSP), as shown in Figure 9-9, unlike the use of CRLs where the list must be downloaded locally for validation and verification. OCSP can be used by the authenticating client itself, through validation requests sent to the respective certificate authorities' OCSP responder to verify the validity of the issued certificate. In more modern deployment scenarios, *OCSP stapling* may be used, which further simplifies the validation flow of an issued certificate.

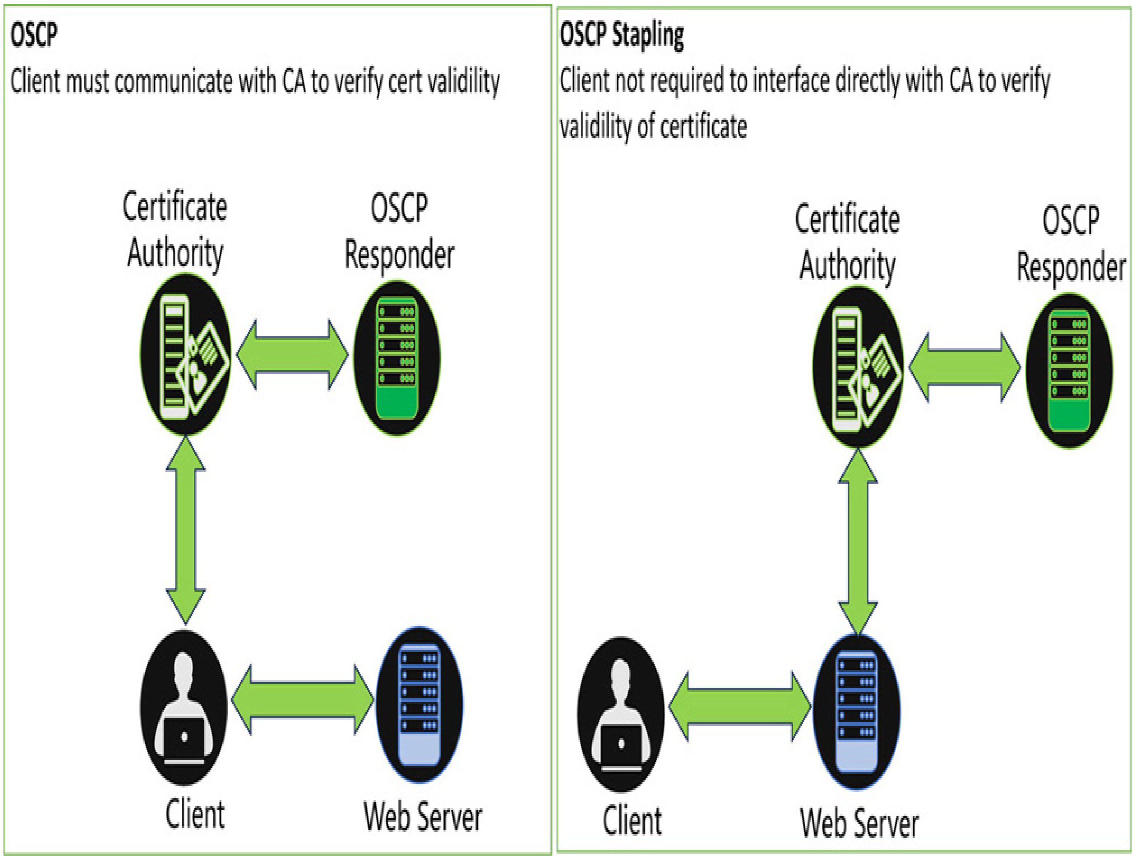

OSCP
Client must communicate with CA to verify cert validility
Certificate Authority
OSCP Responder
Client
Web Server
OSCP Stapling
Client not required to interface directly with CA to verify validility of certificate
Certificate Authority
OSCP Responder
Client
Web Server

Figure 9-9 *OCSP-Based Certificate Validation*

In Cisco software-defined architectures, the use of certificates is very common and considered a best practice. General good operational hygiene involves the deployment of certificates that are signed by a third-party CA. Depending on whether the devices that are using that CA are public facing in the Internet or private devices or servers that may be located in a private enterprise network will dictate whether a public certificate authority such as Symantec or Verisign should be selected, or if a privately managed CA, such as an installation on a Microsoft Windows Server, may be the preferred mode of usage.

REST-API Authentication Methods

Representational state transfer (REST) has become the de facto standard for use within the Internet today. API-driven connectors and architectures that are common across a multitude of vendors and open-source projects and systems use REST-based API logic to allow for a common and standardized abstraction and interface within their platforms. In many architectures, particularly in cloud-based services, there is an API-first mantra, which has opened the door to the deployment of complex architectures using DevOps-based best practices to deploy Infrastructure as Code, which is described in more detail in Chapter 23, "Infrastructure as Code (IaC)."

As is the case with any critical and important architecture, security is a key foundation toward successful deployment.

Authentication methods that are commonly used with REST include

- Basic authentication (Base64 hash of credentials: considered insecure)
- Certificate-based authentication
- Bearer token (issued as a result of a previous authentication)
- API key (not recommended for large-scale deployments)

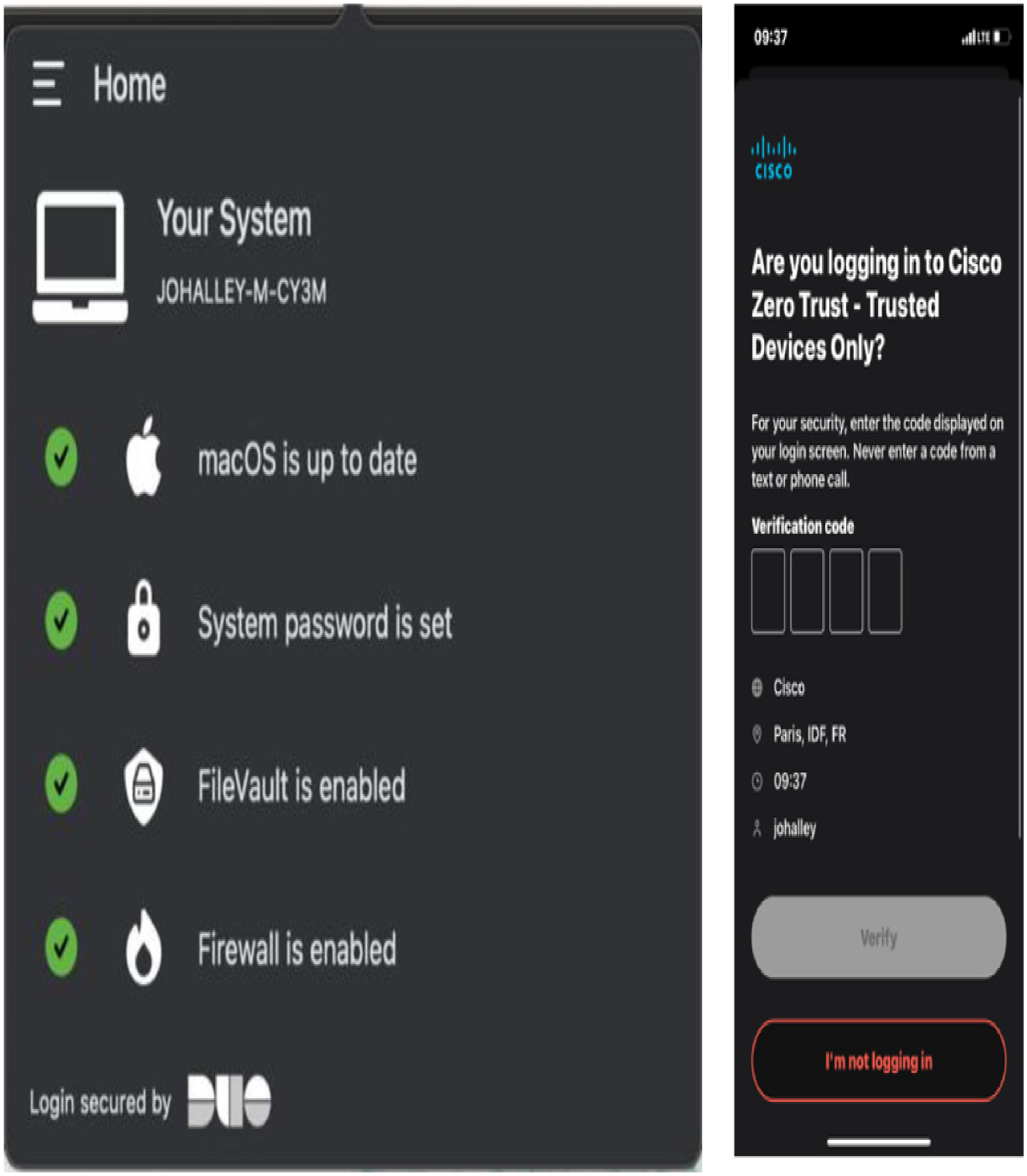
One common misconception is that the OAuth protocol is an authentication protocol; in fact, OAuth only provides authorization.

Multifactor Authentication (MFA)

The use of multifactor authentication (MFA) has become commonplace today, ranging from use within corporate applications to granting network access for users. MFA can be effective in protecting against common attacks such as password spraying. The most basic multifactor authentication methods that many of us have perhaps become accustomed with over years is the use of Short Message Service (SMS). The advantage of this approach in the past was that the process of sending and receiving SMS messages was under the third-party control of the mobile service provider, distancing the one-time key from the user. Typically, to breach or gain access to that data, the attacker needed access to the mobile operator's Short Messaging Service Center (SMSC) server—generally not a straightforward task.

Note that while MFA offers many benefits, we cannot blindly consider that all types and methods used are secure. Breaches have still been observed in scenarios where credential synchronization takes place with cloud services for an already-compromised account in conjunction with secondary attacks through voice phishing (vishing) or MFA fatigue. In these cases, the user is bombarded with authentication requests until they finally approve or accidentally approve the access.

Newer systems today such as Cisco DUO provide not only a viable one-time password to authenticate to a resource but also include a range of verifications to increase the level of security and integrity in the activity. This may include verification of country that the individual is located in at the time of authentication, operating system compliance and verification checks for both systems in use (PC, Mac, Linux) and mobile devices responsible for the challenge, and other checks (see Figure 9-10).



Your System

JOHALLEY-M-CY3M

✓ macOS is up to date

✓ System password is set

✓ FileVault is enabled

✓ Firewall is enabled

Login secured by DUO

Are you logging in to Cisco Zero Trust - Trusted Devices Only?

For your security, enter the code displayed on your login screen. Never enter a code from a text or phone call.

Verification code

Cisco

Paris, IDF, FR

09:37

johalley

Figure 9-10 *System Compliance State Using DUO and Mobile Device One-Time Password Verification*

Network operators can associate network proxies from the MFA systems like DUO to act as the intermediary between systems like the Identity Services Engine (ISE) and Active Directory servers or backend, thereby facilitating MFA login as part of the NAC or VPN onboarding flow when attempting to access a corporate network, as shown in Figure 9-11.

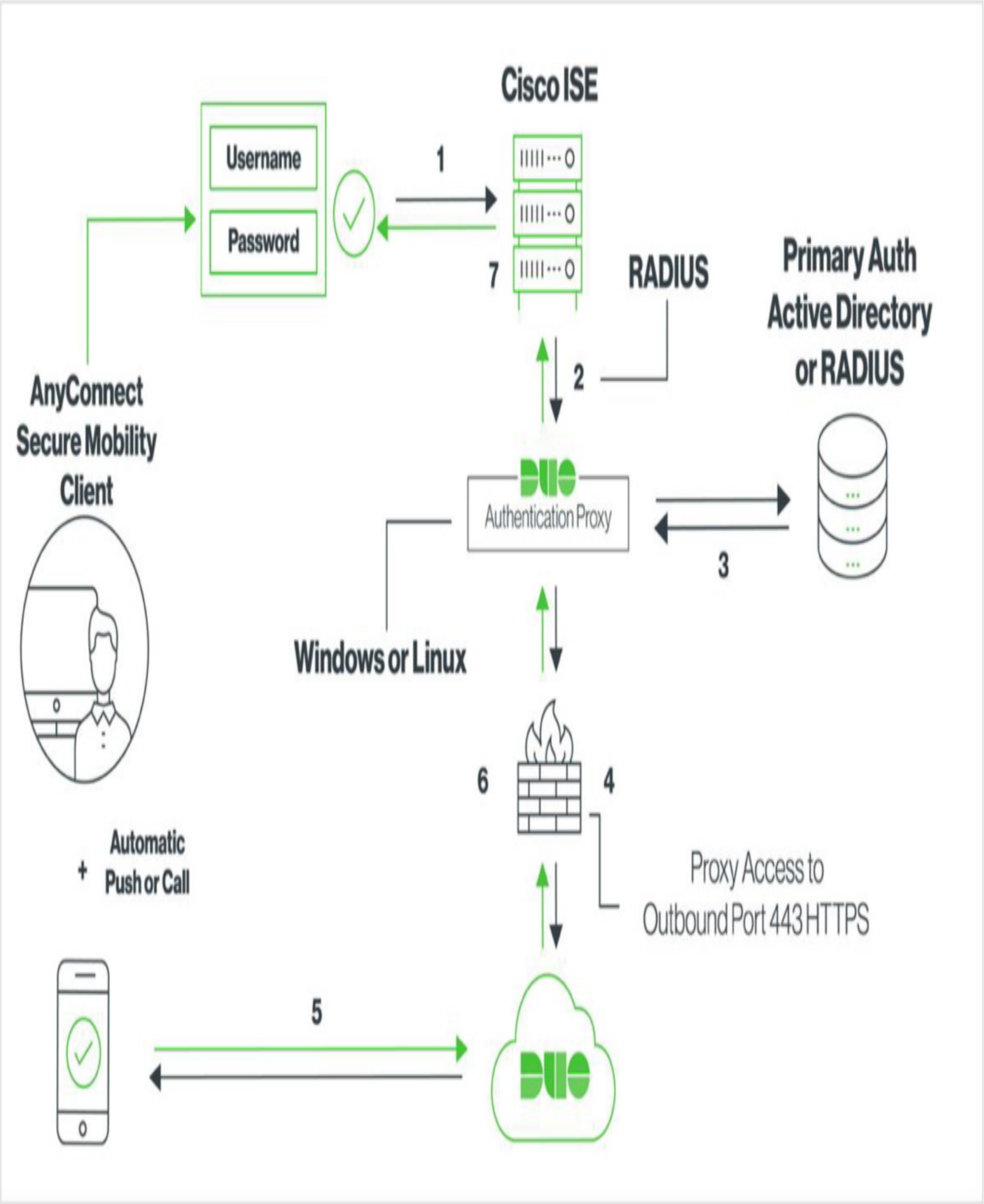


Figure 9-11 *ISE Integration with DUO*

Network Access Control

In legacy IT network environments, wired and wireless devices could gain connectivity to the network without the need for confirming their identity (user or endpoint) or validating credentials. This has evolved to a network access model where access is granted to users or devices by exchanging credentials or x.509 certificate information, sometimes in conjunction with known parameters that may exist through fingerprinting (profiling) devices as a means to further improve the level of trust in the user's or endpoint's identity and as such provide tailored network access.

In a wired network infrastructure, systems can be configured with a priority order to permit user access. For instance, if enterprise authentication using 802.1x is not possible within 20 seconds of the port becoming active, then a secondary authentication mechanism utilizing MAC Authentication Bypass (MAB) may be used. Depending on the network requirements and the types of endpoints that are in use, sometimes, both MAB and dot1x may even execute in parallel. Such options are usually configurable within a network architecture.

MAC Authentication Bypass

As the name rightly describes, MAC Authentication Bypass makes use of the MAC address from an endpoint that can be utilized to permit network access. However, it should be considered as a method of bypassing the authentication method itself. The reason it is considered a bypass action is largely due to the insecurity of this approach, attributed to its lack of encryption and relative ease of spoofing MAC addresses, which can result in exposure for unauthorized access. It is important to note that finding a system's Layer 2 or MAC address is not difficult; in many IoT devices, such as printers or even access points and IP phones, the address is clearly visible on a label, without your even needing to log in to the devices (see Figure 9-12). This address is often used to grant network access, and often even to secure network segments, such as the domains with access to the infrastructure network, such as the global routing table.

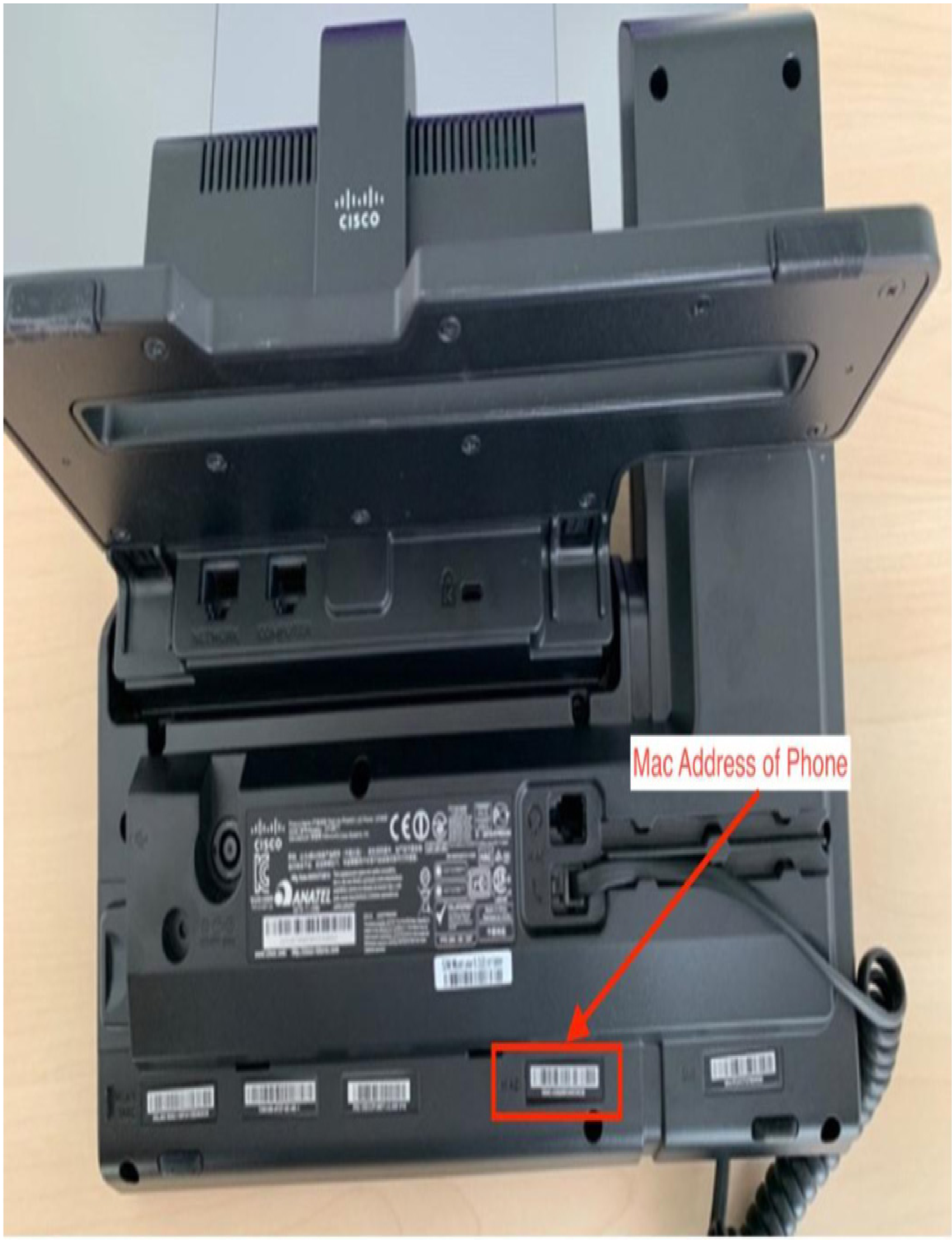
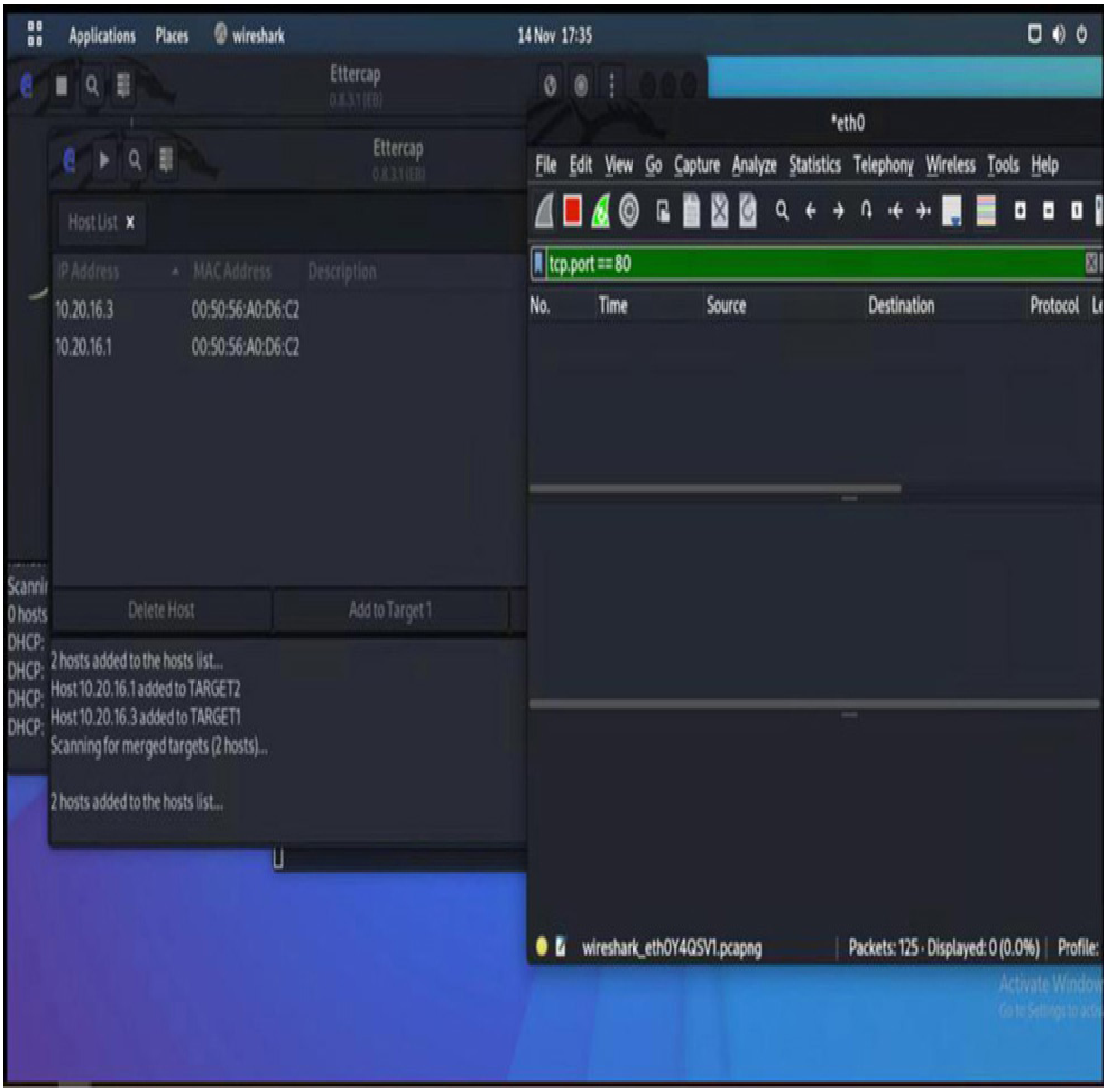


Figure 9-12 *Visible MAC Address on Rear of IP Phone*

The risk that the exposure of this address poses is that if network entry is based on MAB for the wired or wireless network, it is simple for a malicious individual who may be onsite with physical access to copy the address and therefore gain access and rights to the same network segment that the original device had access to by spoofing (impersonating) the Layer 2 address. Once the MAC address of an infrastructure device is known, further exploitation of the network could take place. Attackers potentially could attempt to perform an on-path attack (formerly called a man-in-the-middle [MITM]) attack) using tools like Ettercap, as shown in Figure 9-13, thus gaining access to traffic that is destined from or to the spoofed device. In a scenario where an on-path attack is successful, Wireshark would show RX (receive) and TX (transmit) packets from the compromised hosts.



IP Address	MAC Address	Description
10.20.16.3	00:50:56:A0:D6:C2	
10.20.16.1	00:50:56:A0:D6:C2	

```
2 hosts added to the hosts list...
Host 10.20.16.1 added to TARGET2
Host 10.20.16.3 added to TARGET1
Scanning for merged targets (2 hosts)...

2 hosts added to the hosts list...
```

```
tcp.port == 80
```

```
wireshark_eth0Y4QSV1.pcapng | Packets: 125 · Displayed: 0 (0.0%) | Profile:
```

Figure 9-13 *Ettercap Being Used to Attempt ARP Injection Exploit*

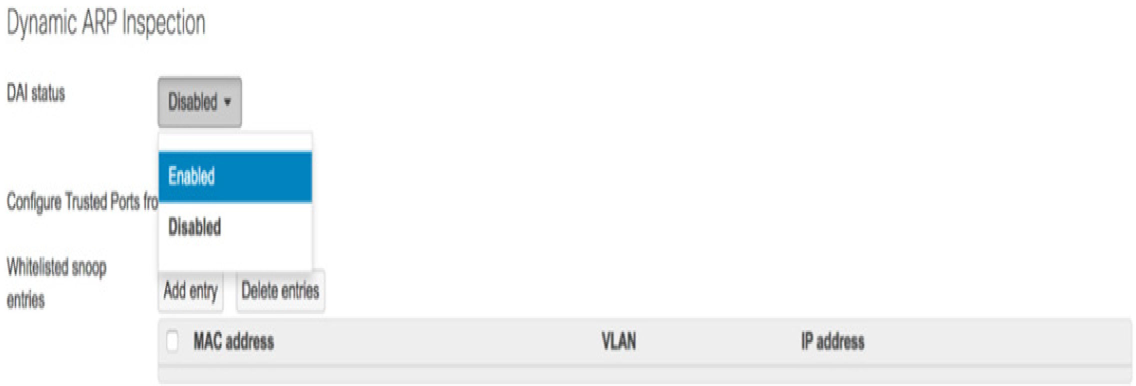
In some operating systems, the ability to change the Layer 2 MAC address to something completely different that could be used maliciously is very easy. Such an example is shown from an OSX UNIX operating system in Example 9-2. Note that the exact syntax and interface naming convention used may vary slightly from one UNIX/Linux distribution to another.

Example 9-2 *MAC Address Spoofing in OSX*

```
ifconfig en0 | grep ether <<< Verify existing address bound to in

ether 88:66:5a:3d:39:21 <<< Current MAC Address

sudo ifconfig en0 ether CA:FE:CA:FE:CA:FE       <<< Changing (spo
```

While this sort of attack represents a potential security concern, various safeguards can be considered for use that may help in mitigating MAC spoofing exploits being used within a NAC-based network.

Software-defined network architectures such as Cisco SD-Access and Cisco Meraki can utilize an advanced device-tracking policy, as shown in Example 9-3, or dynamic ARP inspection, as shown in Figure 9-14, to avoid potential attack vectors, such as on-path (aka MITM) attacks.



Dynamic ARP Inspection

DAI status: Disabled

Enabled

Disabled

Configure Trusted Ports fro

Whitelisted snoop entries

MAC address	VLAN	IP address

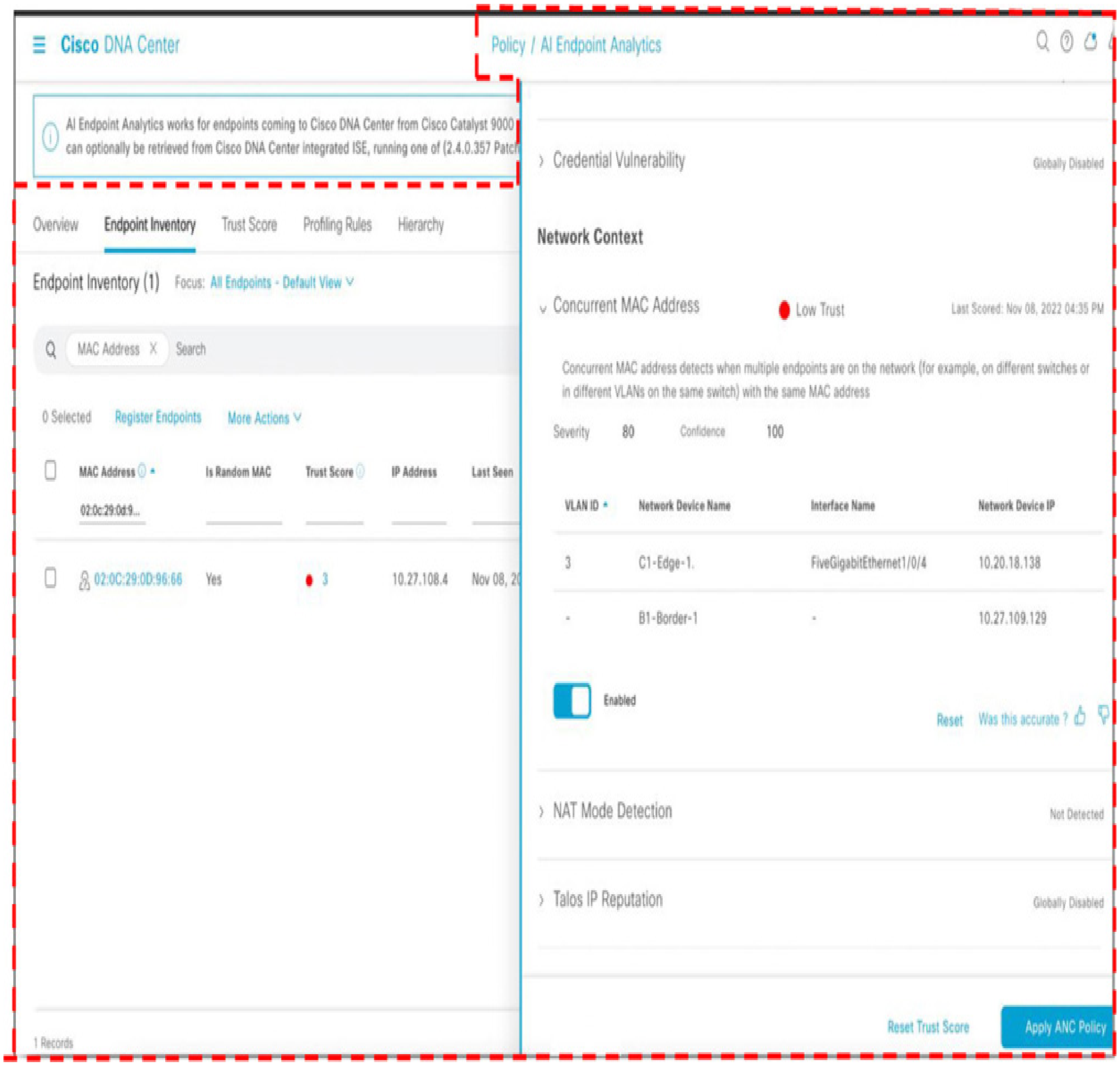
Figure 9-14 *Dynamic ARP Inspection (Meraki)*

Example 9-3 provides an overview of an IOS XE interface configuration from a Cisco Catalyst series switch and the IPDT policy. This configuration shows the relevant base configuration associated with the default VLAN and port settings, together with a mapping to the IP Device Tracking policy, which applies security features to protect against activities such as MAC spoofing and IP theft.

Example 9-3 *Device Tracking Policy Preventing Localized MAC Spoofing in SD-Access*

```
interface FiveGigabitEthernet1/0/4
 switchport access vlan 530
 switchport mode access
 device-tracking attach-policy IPDT_POLICY <<< ARP protection via
 ip flow monitor dnacmonitor input
 ip flow monitor dnacmonitor output
 load-interval 30
 ipv6 flow monitor dnacmonitor_v6 input
 ipv6 flow monitor dnacmonitor_v6 output
 access-session inherit disable interface-template-sticky
 access-session inherit disable autoconf
 no macro auto processing
 spanning-tree portfast
 spanning-tree bpduguard enable
 ip nbar protocol-discovery
end
Switch# show run all | section IPDT_POLICY
device-tracking policy IPDT_POLICY
 security-level guard
 device-role node
 medium-type-wireless
 no data-glean
 no destination-glean
 protocol ndp
 protocol dhcp6
 protocol arp
 protocol dhcp4
 tracking enable reachable-lifetime 300
```

Performing an on-path attack typically requires a level of localized Layer 2 access to the network, hence allowing traffic to be diverted to the attacker. Still, having the Layer 2 address can allow breaches through unauthorized network access to take place in disparate and remote locations on the network, potentially even in different sites or countries if the authentication policy that was deployed on the authentication architecture (such as a RADIUS server) does not limit MAB entries to a particular location. To avoid such scenarios, Cisco Catalyst Center can use its AI Endpoint Analytics feature, as shown in Figure 9-15, to detect concurrent MAC address usage that may appear within the estate that is under the systems control.



AI Endpoint Analytics works for endpoints coming to Cisco DNA Center from Cisco Catalyst 9000
can optionally be retrieved from Cisco DNA Center integrated ISE, running one of (2.4.0.357 Patch

☐	MAC Address	Is Random MAC	Trust Score	IP Address	Last Seen
	02:0c:29:0d:9...				
☐	02:0C:29:0D:96:66	Yes	● 3	10.27.108.4	Nov 08, 20

Last Scored: Nov 08, 2022 04:35 PM

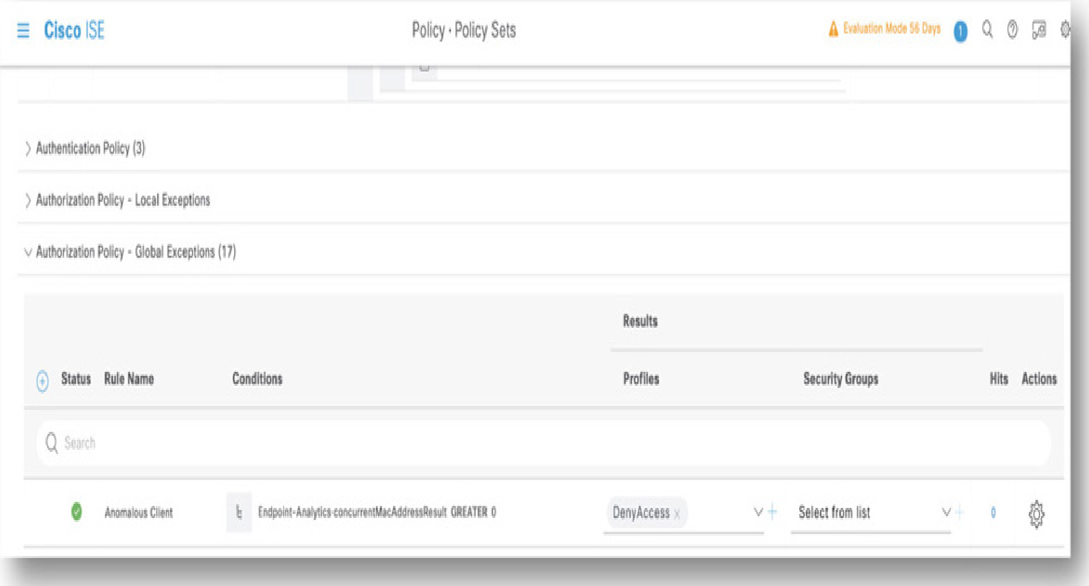
Concurrent MAC address detects when multiple endpoints are on the network (for example, on different switches or in different VLANs on the same switch) with the same MAC address

Severity 80 Confidence 100

VLAN ID	Network Device Name	Interface Name	Network Device IP
3	C1-Edge-1.	FiveGigabitEthernet1/0/4	10.20.18.138
-	B1-Border-1	-	10.27.109.129

Figure 9-15 *Concurrent MAC Detection in Cisco Catalyst Center*

When using endpoint analytics attributes in conjunction with ISE policy, you can configure conditional rejection rules, as shown in Figure 9-16, to execute when spoofed MAC addresses are identified within the deployment.


Cisco ISE
Policy · Policy Sets
Evaluation Mode 56 Days
Authentication Policy (3)
Authorization Policy - Local Exceptions
Authorization Policy - Global Exceptions (17)
Search

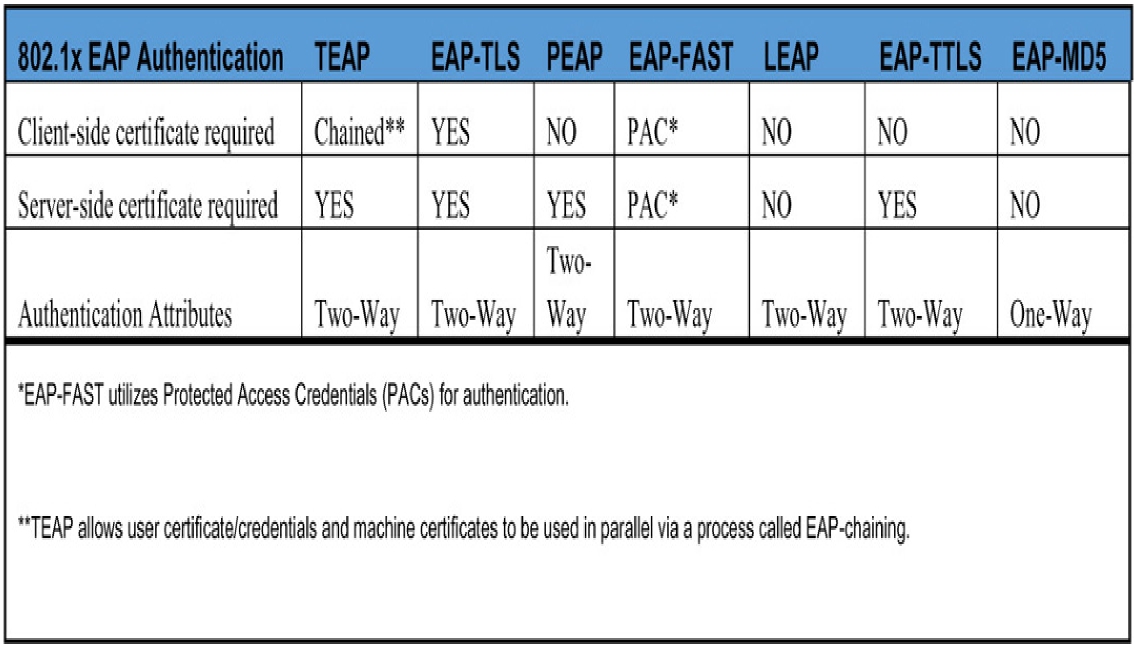
			Results			
Status	Rule Name	Conditions	Profiles	Security Groups	Hits	Actions
✓	Anomalous Client	Endpoint-Analytics-concurrentMacAddressResult GREATER 0	DenyAccess	Select from list	0	⚙

Figure 9-16 *Conditional Rejection Rules in ISE Authorization Policy for Concurrent MAC Address Detection*

802.1x (Network Authentication)

The standard that is used for passing Extensible Authentication Protocol (EAP) over wired and wireless local area networks is 802.1x. In corporate networks and environments, the use of network authentication methods that utilize 802.1x with centralized server–based authentication is expected. This authentication method involves a scenario where the client device attempts to authenticate to the network using the EAP framework. Simply turning on a dot1x-based EAP method of authentication, however, does not necessarily mean that the method is very secure. Over the years, many EAP methods have come and gone. In terms of their level of security, methods such as LEAP and EAP-MD5 were once considered viable protocols for authentication; however, as years went by, they were later identified as being easily prone to exploit and attack. Table 9-2 provides an overview of common EAP methods and their respective differences.

Table 9-2 *Common EAP Methods*



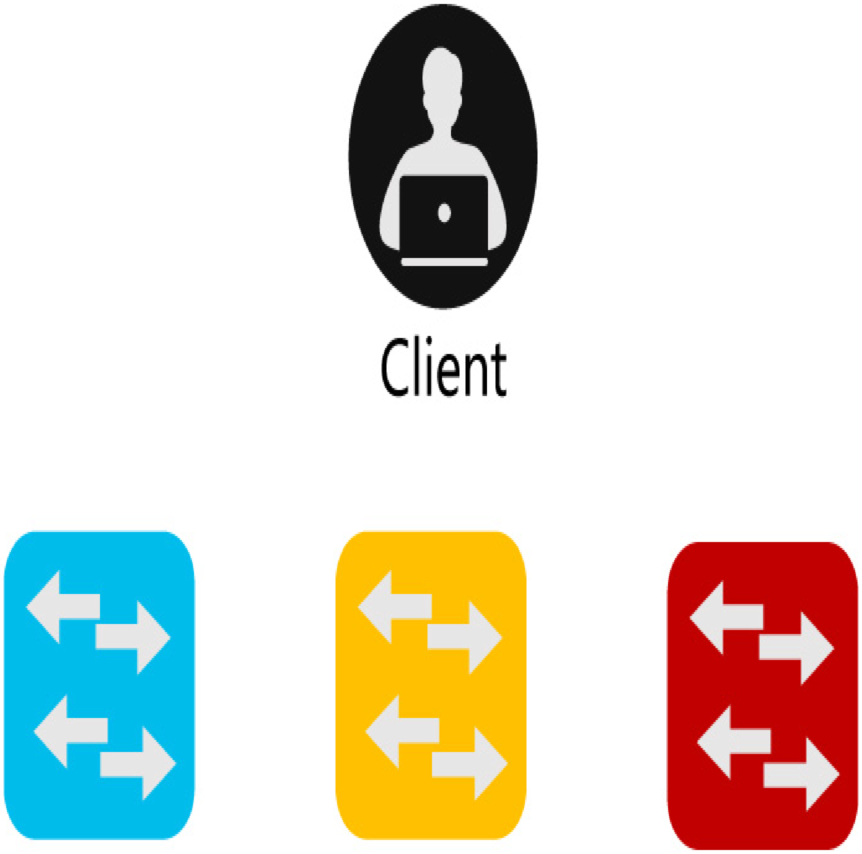
802.1x EAP Authentication	TEAP	EAP-TLS	PEAP	EAP-FAST	LEAP	EAP-TTLS	EAP-MD5
Client-side certificate required	Chained**	YES	NO	PAC*	NO	NO	NO
Server-side certificate required	YES	YES	YES	PAC*	NO	YES	NO
Authentication Attributes	Two-Way	Two-Way	Two-Way	Two-Way	Two-Way	Two-Way	One-Way

*EAP-FAST utilizes Protected Access Credentials (PACs) for authentication.

**TEAP allows user certificate/credentials and machine certificates to be used in parallel via a process called EAP-chaining.

In most corporations today, EAP-TLS and EAP-TEAP are considered viable methods for secure authentication where both client- and server-side certificates are mutually validated. It is worth mentioning that EAP-TLS and EAP-TEAP are some of the most secure authentication methods. However, in a wired access environment, once the authentication process has completed and the access port has move to the “authorized” state, the authentication methods that are employed do not provide authentication on a packet-by-packet basis, nor do they protect traffic flowing from the endpoint to the network access server in any way. Therefore, authentication may be bypassed using a hardware-based attack consisting in inserting a device capable of EAPoL bridging in between the endpoint and the access switch. To mitigate these types of scenarios, a revision of the 802.1x standard (802.1X-2010) introduced MACsec, which provides Layer 2 encryption capabilities on a hop-by-hop basis as well as packet-by-packet integrity checks.

When you’re considering how to improve the overall security of a supplicant’s connectivity to a network, augmenting the insecurity of MAB-only-based authorization with further attributes that can be derived via profiling, such as DHCP, CDP, and other input, you can achieve an improved level of security. However, this approach remains significantly insecure in contrast to the use of EAP methods, such as EAP-TLS and TEAP, as shown in Figure 9-17.

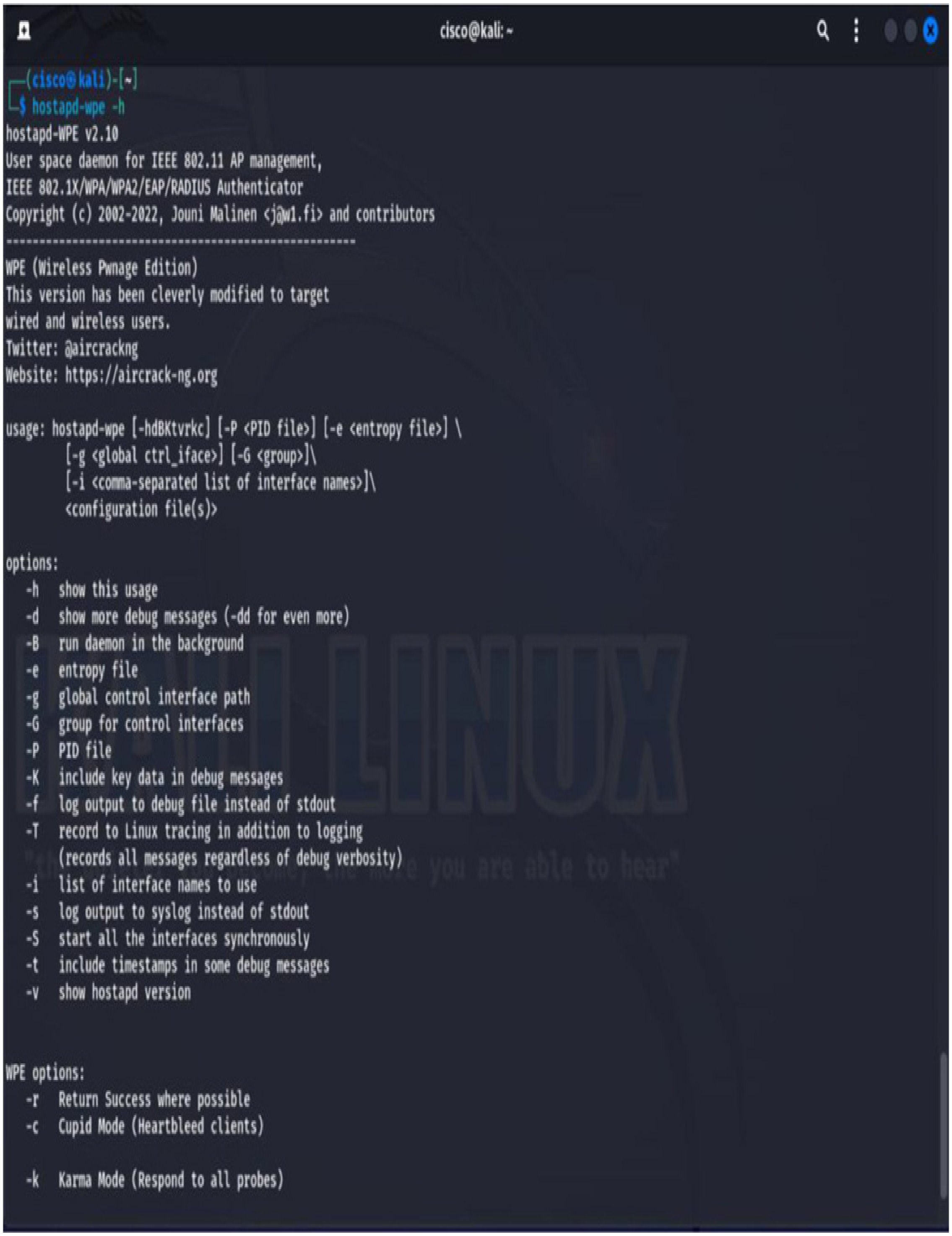


Secure	Insecure	Very Insecure
EAP Methods	Profiling	MAB
EAP-TLS	DHCP	cafe.cafe.cafe
TEAP	CDP	
Etc..	LLDP	
	CBAR	
	OUI	
	NMAP	
	Etc..	

Figure 9-17 *Security of Authentication Methods*

In the past, many network deployments used EAP-PEAP with MSCHAPv2 because it would link up with LDAP or Active Directory deployments, allowing users to leverage their corporate username and passwords without needing to provision endpoints with certificates. While this approach sounds straightforward, it very quickly became a simplistic means for attackers to infiltrate the network due to vulnerabilities that reduced the security of MS-CHAPv2 to a single DES encryption (2^56), regardless of the password length (further described in MS Security Advisory 2743314).

Two tools provided a simple means to start harvesting username and password hash. The first tool is called hostapd-wpe, which allows attackers to spoof an access point, creating a “rogue AP,” and spoof the SSID from the corporate network. Attackers would often even sit outside corporate offices running their rogue devices in areas of low or no coverage from the corporate SSID to sway employees to log in. As users attempted to log in using their EAP credentials, the password challenges were saved to allow for later attempts to uncover the passwords that users leveraged. Figure 9-18 shows the rich set of capabilities that the hostapd-wpe tool has to allow malicious users to attempt to exploit the network using this method.

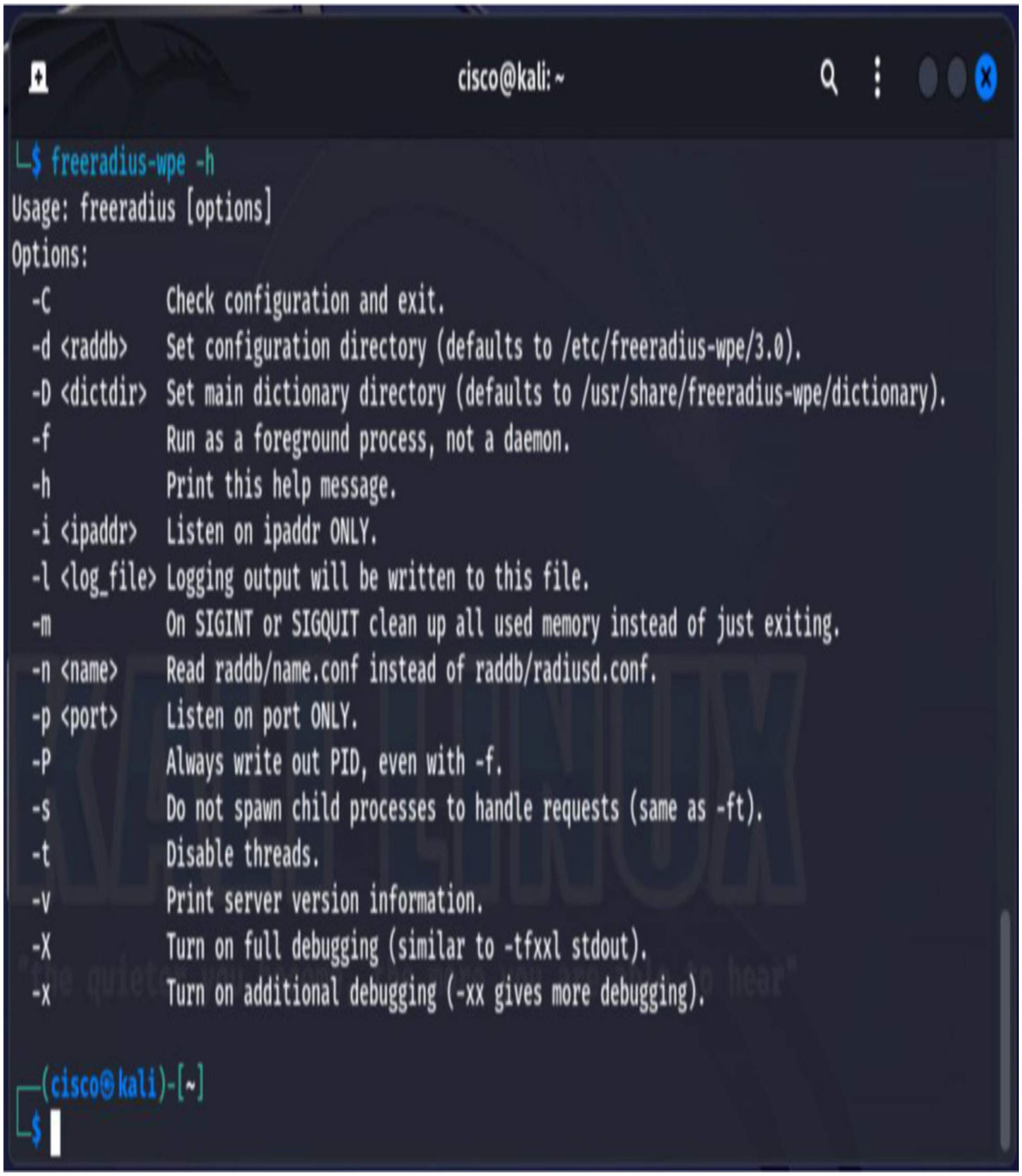


```
┌──(cisco㉿kali)-[~]
└─$ hostapd-wpe -h
hostapd-WPE v2.10
User space daemon for IEEE 802.11 AP management,
IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator
Copyright (c) 2002-2022, Jouni Malinen <j@w1.fi> and contributors
-----------------------------------------------------
WPE (Wireless Pwnage Edition)
This version has been cleverly modified to target
wired and wireless users.
Twitter: @aircrackng
Website: https://aircrack-ng.org

usage: hostapd-wpe [-hdBKtvrkc] [-P <PID file>] [-e <entropy file>] \
         [-g <global ctrl_iface>] [-G <group>]\
         [-i <comma-separated list of interface names>]\
         <configuration file(s)>

options:
   -h   show this usage
   -d   show more debug messages (-dd for even more)
   -B   run daemon in the background
   -e   entropy file
   -g   global control interface path
   -G   group for control interfaces
   -P   PID file
   -K   include key data in debug messages
   -f   log output to debug file instead of stdout
   -T   record to Linux tracing in addition to logging
        (records all messages regardless of debug verbosity)
   -i   list of interface names to use
   -s   log output to syslog instead of stdout
   -S   start all the interfaces synchronously
   -t   include timestamps in some debug messages
   -v   show hostapd version


WPE options:
   -r   Return Success where possible
   -c   Cupid Mode (Heartbleed clients)

   -k   Karma Mode (Respond to all probes)
```

Figure 9-18 *hostapd-wpe Wireless Exploit Tool*

Another common tool that is often used for the same purpose is FreeRADIUS-WPE (Wireless Pwnage Edition), which is an instrumented version of the FreeRADIUS open-source RADIUS server (see Figure 9-19).

Similar to hostapd-WPE, the primary objective of this FreeRADIUS system is to collect authentication credentials from users, which can subsequently be utilized to gain unauthorized access to the network or the client’s assets, particularly in situations where the same username and password are commonly employed.



```
└─$ freeradius-wpe -h
Usage: freeradius [options]
Options:
  -C            Check configuration and exit.
  -d <raddb>    Set configuration directory (defaults to /etc/freeradius-wpe/3.0).
  -D <dictdir>  Set main dictionary directory (defaults to /usr/share/freeradius-wpe/dictionary).
  -f            Run as a foreground process, not a daemon.
  -h            Print this help message.
  -i <ipaddr>   Listen on ipaddr ONLY.
  -l <log_file> Logging output will be written to this file.
  -m            On SIGINT or SIGQUIT clean up all used memory instead of just exiting.
  -n <name>     Read raddb/name.conf instead of raddb/radiusd.conf.
  -p <port>     Listen on port ONLY.
  -P            Always write out PID, even with -f.
  -s            Do not spawn child processes to handle requests (same as -ft).
  -t            Disable threads.
  -v            Print server version information.
  -X            Turn on full debugging (similar to -tfxxl stdout).
  -x            Turn on additional debugging (-xx gives more debugging).

┌──(cisco㉿kali)-[~]
└─$ 
```

Figure 9-19 *FreeRADIUS-WPE Instrumented RADIUS Server*

Although there are tools in the field that can simplify an attacker's means to breach a corporate environment, fortunately, there are just as many tools and capabilities that impose the right guardrails around access to the network and resources to create the right levels of boundaries to secure network and resource access.

Authorization

Whereas authentication is focused on verifying the identity of an individual or a device or system, authorization is focused on which privileges or level of access or capability the respective individual or device may be provided. In the context of application access on the Internet or to corporate systems, protocols such as OAUTH 2.0 are commonly used. In IP-based networks, TACACS+ and RADIUS are most used to achieve this task.

Following are common examples of such authorization capabilities applied via TACACS+ or RADIUS:

- IPv4/IPv6 network access via statically configured access-control list
- Dynamic IPv4/IPv6 ACL (only possible on wired and FlexConnect)
- VLAN allocation
- VLAN group allocation
- Quality of service rate limit
- Quality of service DSCP marking
- Quality of service WMM marking (Wi-Fi)
- Quality of service policy map application
- Quality of service AVC
- RADIUS NAC user role
- Time-based session
- TrustSec/Adaptive policy security group tag

- Interface template configuration
- Selective allocation of AAA server
- Posture triggered script execution
- Security triggered change of access rights (Cisco Secure Network Analytics (SNA))
- GUI access to systems (Catalyst Center, Identity Services Engine, Meraki Dashboard)

Note

Although it would be possible to write an entire book just on the use of all the different dynamic options that are available, you can see the application of some of these options in Example 9-4.

As can be seen from the preceding list of parameters, key and critical options that would normally be required to configure network ports and profiles throughout an architecture can, in fact, be configured centrally via a RADIUS server, such as ISE. These configurations enable you to shift away from the legacy approach of configuring a network on a port-by-port or even SSID-by-SSID basis to a configuration where all policies, rules, capabilities, and operational limitations are defined in a central location. Such an approach creates a much more simplistic way to ensure standards are applied throughout the network and to allow for updates and changes to be tested and applied in disparate locations via the use of policy.

Authorization capabilities are typically configured through the use of server-driven policy, such as RADIUS. In wired and wireless networks, you can find such an applied policy listed under “Server Policies” when checking the connected users’ access session for wired or wireless client details.

Under the Server Policies output shown in Example 9-4, you can see an overview of the different dynamic radius policies applied to the RADIUS session 10301B0A000001B8F082D4EE:

- Dynamic VLAN allocation through the Vlan Group entry

- Dynamic IP access list called xACSACLx-IP-Posture_DACL-633f0c5b with the ACS ACL entry
- A URL deployed for supplicant redirection to a Guest or Posture portal with the URL Redirect entry
- A Redirection ACL applied with the URL Redirect ACL entry to specify which traffic redirects to the configured portal and which traffic is discarded
- Security group tag allocation is also applied via the SGT Value entry

Example 9-4 *Server Policies Dynamically Sent to Authenticator (Switch) Terminating Supplicant dot1x Session*

```
B2-Ext-1#show access-session mac   0050.56a0.500a details
            Interface:  GigabitEthernet1/0/8
               IIF-ID:  0x10E99F1D
          MAC Address:  0050.56a0.500a
         IPv6 Address:  Unknown
         IPv4 Address:  10.27.44.9
            User-Name:  Cisco Press
          Device-type:  Microsoft-Workstation
          Device-name:  DESKTOP-80I04F5
               Status:  Authorized
               Domain:  DATA
       Oper host mode:  multi-auth
     Oper control dir:  both
      Session timeout:  N/A
  Acct update timeout:  172800s (local), Remaining: 172794s
    Common Session ID:  10301B0A000001B8F082D4EE
      Acct Session ID:  0x0000446b
               Handle:  0xdd00018e
       Current Policy:  PMAP_DefaultWiredDot1xClosedAuth_1X_MAB
Server Policies:
           Vlan Group:  Vlan: 303 <<< Dynamic VLAN Allocation
```

```
                ACS ACL: xACSACLx-IP-Posture_DACL-633f0c5b << Dynam
       URL Redirect ACL: ACL_POSTURE_REDIRECT << Redirection ACL
              SGT Value: 2023 << Security Group Tag
           URL Redirect:
https://ise3.net.cisco.com:8443/portal/gateway?sessionId=10301B0A
b26f-8b022bba10e7andaction=cppandtoken=8bfa6ce2f44c630007b46805a8
Method status list:
       Method           State
        dot1x           Authc Success
```

Dynamic Change of Authorization (CoA)

Dynamic policy provides a rapid, scalable, and robust means to manage modern network architectures, moving away from the transitional and static methods of configuring architectures of the past on a port-by-port basis. However, what if something happens and the state of a client or endpoint needs to change and adopt policies or rules that may differ based on more elaborate and involved conditions? The simplest example of a change of dynamic authorization state would be a wireless guest portal. During initial connectivity until the acceptable use policy (AUP) or user details are entered, Internet access is generally restricted. After these details are entered, the client can finally access the Internet. In the case of the wireless guest user, the policy would normally indicate the removal of a restrictive access list that is in place during the initial connection phase, prior to AUP acceptance. In other scenarios, however, a change of authorization (CoA) could be triggered for a variety of different reasons, such as a mobile device management system triggering a containment action, or increased or reduced bandwidth being applied through a quality of service (QoS) policy. Example 9-4 shows a number of Server Policies that are derived from the RADIUS servers' authorization configurations. In this example, the VLAN number, a dynamic access list, a redirection access list for the use of posture validation, a security group tag, and a redirection URL are all pushed dynamically as part of the client's session. Further examples of server policies that could be applied are time-based policies for resource access or session limits, compliance-based policies that could be applied through consulting an MDM or posture server, or a QoS-based policy such as rate limits.

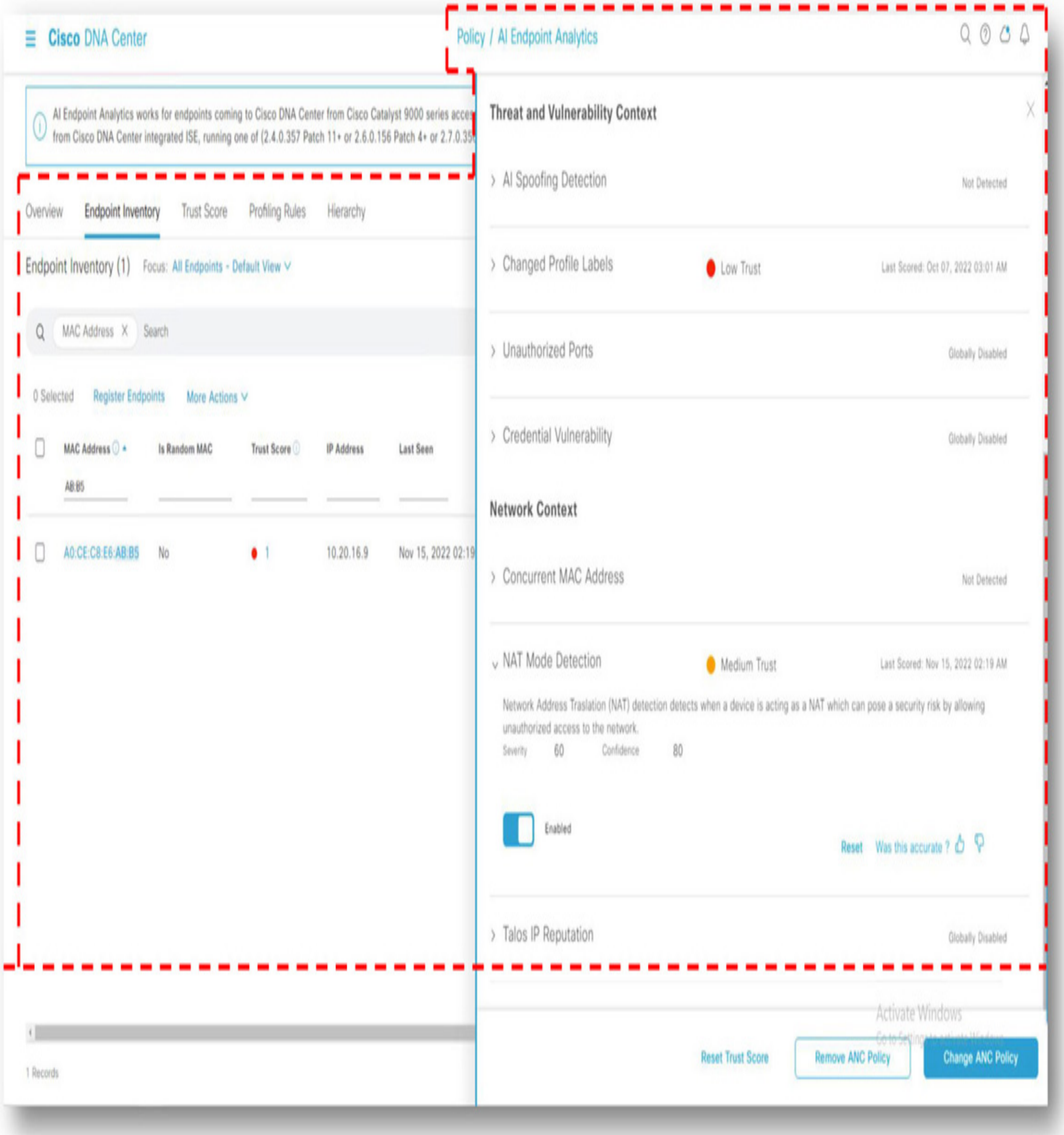
While the number of dynamic attributes may vary from deployment to deployment, the more that you can harness from dynamic capabilities and the less reliance there is on static configurations, the more robust and efficient you can make the deployment.

Identifying and Mitigating Risks of Unmaintained Virtual Machines in Network Access Control Deployments

Although it may be possible to ensure that PC, Mac, or Linux devices are running the correct patches and antivirus software to be deployed on a corporate network, what methods can be used for virtual machines (VMs)? There are many types of client software available to run a virtual system, from VMware Workstation, Fusion, Parallels, and Virtual Box, among others, that offer the ability to run a VM. While this capability is often of benefit for users who require different operating systems, without the need to have many physical machines, if the hypervisor on a client workstation is operating in NAT mode, often it will obscure the fact that a VM is running from the view of the network, leading network operators to assume that all systems on their network are running the right levels of compliance validations through systems like Cisco Secure Client and mobile device management systems such as JAMF and Microsoft Intune. However, in reality, a virtual machine that has the same level of network access as the main system may be infected with a virus or may not be staying up-to-date with its OS patches.

Ensuring that virtual machines maintain compliance with corporate network security standards is a challenge when VMs can be obscured by their host hypervisor’s NAT mode. Fortunately, when using Catalyst Center, you can avoid this particular scenario by activating AI Endpoint Analytics’ NAT Mode Detection, as depicted in Figure 9-20. This feature allows network components to be configured with controller-based application recognition (CBAR), allowing the identification of select packet signatures and sending this information onward to the Endpoint Analytics service within Catalyst

Center. Through the communication of anomaly events to ISE, NATed devices that have been detected on a broad range of Cisco Catalyst 9000 switches and Cisco catalyst wireless controllers can subsequently be configured using policies that allow permission or denial based enforcement actions to take place.



AI Endpoint Analytics works for endpoints coming to Cisco DNA Center from Cisco Catalyst 9000 series acces
from Cisco DNA Center integrated ISE, running one of (2.4.0.357 Patch 11+ or 2.6.0.156 Patch 4+ or 2.7.0.35

Endpoint Inventory (1) Focus: All Endpoints - Default View ∨

□	MAC Address •	Is Random MAC	Trust Score	IP Address	Last Seen
	AB:B5				
□	A0:CE:C8:E6:AB:B5	No	● 1	10.20.16.9	Nov 15, 2022 02:19

Threat and Vulnerability Context

› AI Spoofing Detection — Not Detected

› Changed Profile Labels — ● Low Trust — Last Scored: Oct 07, 2022 03:01 AM

› Unauthorized Ports — Globally Disabled

› Credential Vulnerability — Globally Disabled

Network Context

› Concurrent MAC Address — Not Detected

∨ NAT Mode Detection — ● Medium Trust — Last Scored: Nov 15, 2022 02:19 AM

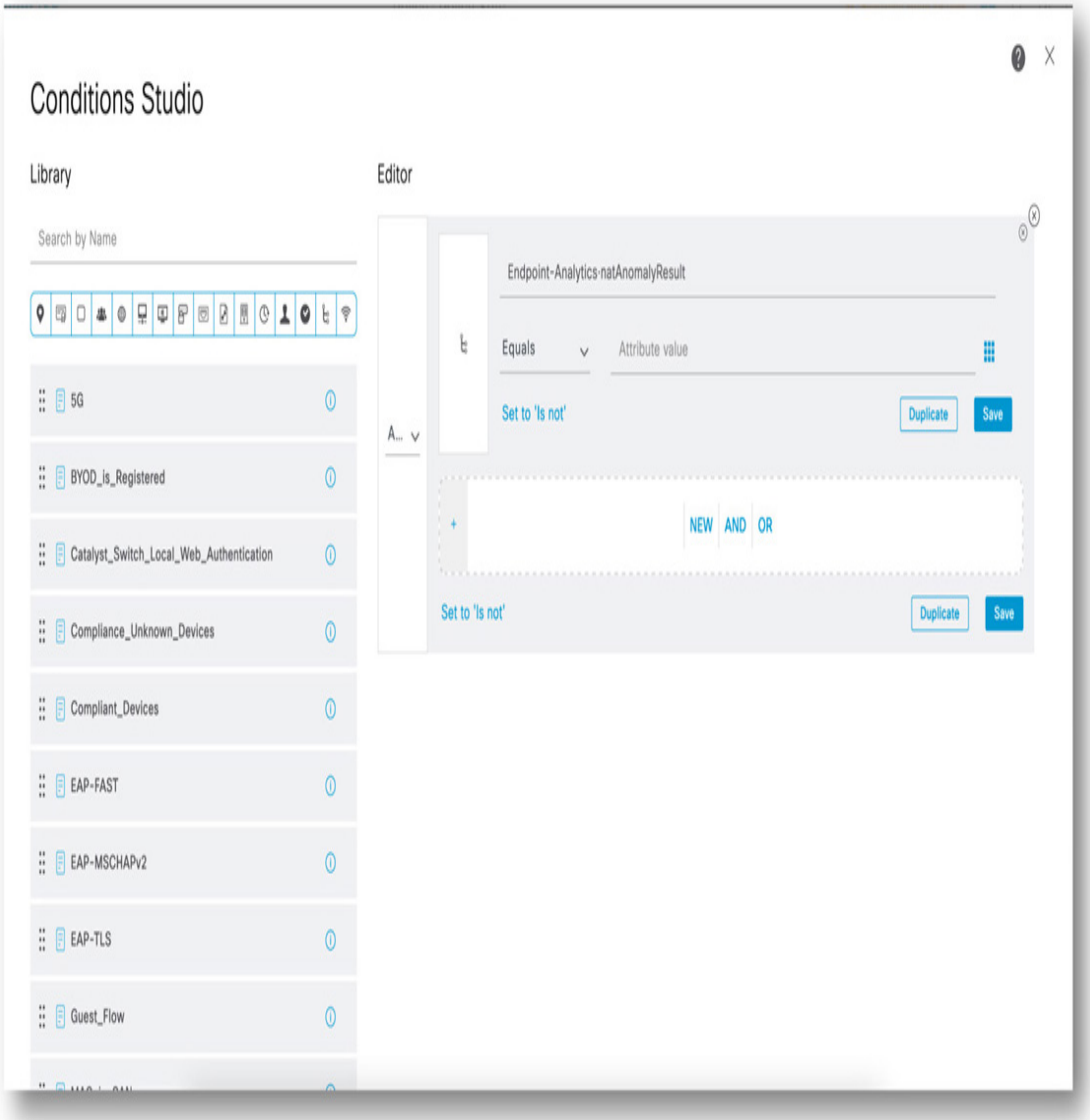
Network Address Traslation (NAT) detection detects when a device is acting as a NAT which can pose a security risk by allowing unauthorized access to the network.

Severity 60 Confidence 80

› Talos IP Reputation — Globally Disabled

Figure 9-20 *NAT Mode Detection in AI Endpoint Analytics*

When they are leveraged in conjunction with a RADIUS policy in ISE that matches a NAT anomaly result (as shown in Figure 9-21), endpoints that are running devices that perform Network Address Translation through their hypervisor can be blocked from access to the network.



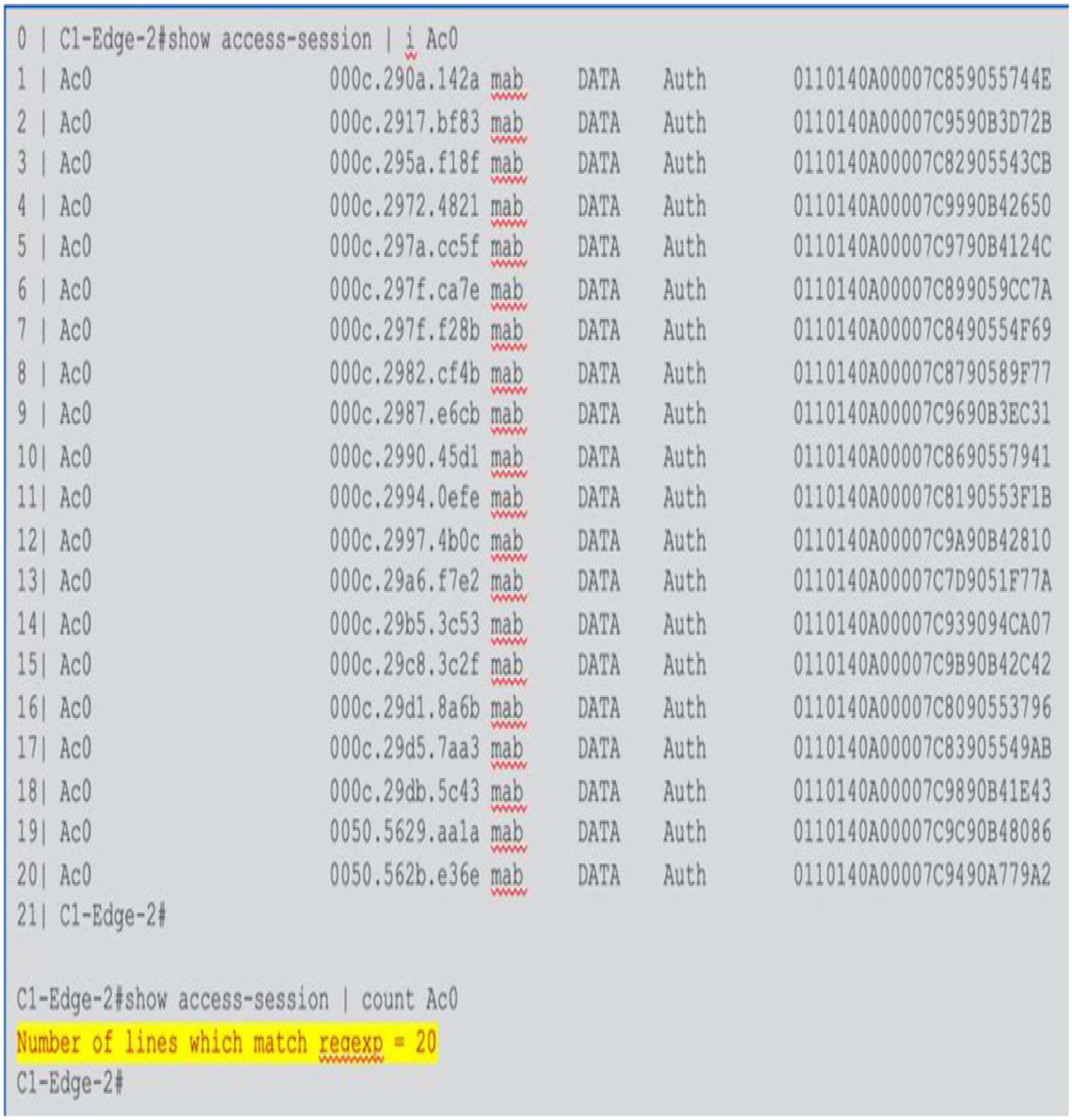
Endpoint-Analytics-natAnomalyResult

Equals Attribute value

Figure 9-21 *NAT Anomaly Condition Configured in ISE*

Taking this approach allows you to force a networkwide policy such that when corporate endpoints use hypervisors like VMware Workstation or parallels, they must operate in bridge mode. How exactly is this any better? The answer to that question is visibility. When virtual machines bridge to the network instead of performing NAT, wired clients can have their VMs leverage security features such as dot1x to authenticate against the network and even use more advanced functionality like posture and TrustSec. With such capabilities available, you can restrict network access to different levels, based on the conditions associated with the main host system and the respective VMs.

For wireless endpoints, the use of the wireless bridged virtual machine feature in Catalyst Center with SD-Access can also allow VMs on a wireless client to be exposed to the network in bridged mode. In this way, their bridged VMs can perform independent authentication and authorization and session profiling activities against switches that are terminating their access point tunnel connection, as shown in Figure 9-22. As is the case with wired connections, for wireless connections, advanced functionality such as TrustSec, Posture, and Redirection can be applied to wireless bridged VMs; however, authentication must be limited to MAB due to restrictions in the passage of EAP packets over the wireless medium.



```
C1-Edge-2#show access-session | i Ac0
Ac0                  000c.290a.142a mab    DATA   Auth       0110140A00007C859055744E
Ac0                  000c.2917.bf83 mab    DATA   Auth       0110140A00007C9590B3D72B
Ac0                  000c.295a.f18f mab    DATA   Auth       0110140A00007C82905543CB
Ac0                  000c.2972.4821 mab    DATA   Auth       0110140A00007C9990B42650
Ac0                  000c.297a.cc5f mab    DATA   Auth       0110140A00007C9790B4124C
Ac0                  000c.297f.ca7e mab    DATA   Auth       0110140A00007C899059CC7A
Ac0                  000c.297f.f28b mab    DATA   Auth       0110140A00007C8490554F69
Ac0                  000c.2982.cf4b mab    DATA   Auth       0110140A00007C8790589F77
Ac0                  000c.2987.e6cb mab    DATA   Auth       0110140A00007C9690B3EC31
Ac0                  000c.2990.45d1 mab    DATA   Auth       0110140A00007C8690557941
Ac0                  000c.2994.0efe mab    DATA   Auth       0110140A00007C8190553F1B
Ac0                  000c.2997.4b0c mab    DATA   Auth       0110140A00007C9A90B42810
Ac0                  000c.29a6.f7e2 mab    DATA   Auth       0110140A00007C7D9051F77A
Ac0                  000c.29b5.3c53 mab    DATA   Auth       0110140A00007C939094CA07
Ac0                  000c.29c8.3c2f mab    DATA   Auth       0110140A00007C9B90B42C42
Ac0                  000c.29d1.8a6b mab    DATA   Auth       0110140A00007C8090553796
Ac0                  000c.29d5.7aa3 mab    DATA   Auth       0110140A00007C83905549AB
Ac0                  000c.29db.5c43 mab    DATA   Auth       0110140A00007C9890B41E43
Ac0                  0050.5629.aa1a mab    DATA   Auth       0110140A00007C9C90B48086
Ac0                  0050.562b.e36e mab    DATA   Auth       0110140A00007C9490A779A2
C1-Edge-2#

C1-Edge-2#show access-session | count Ac0
Number of lines which match regexp = 20
C1-Edge-2#
```

Figure 9-22 *Wireless Bridged VM Example*

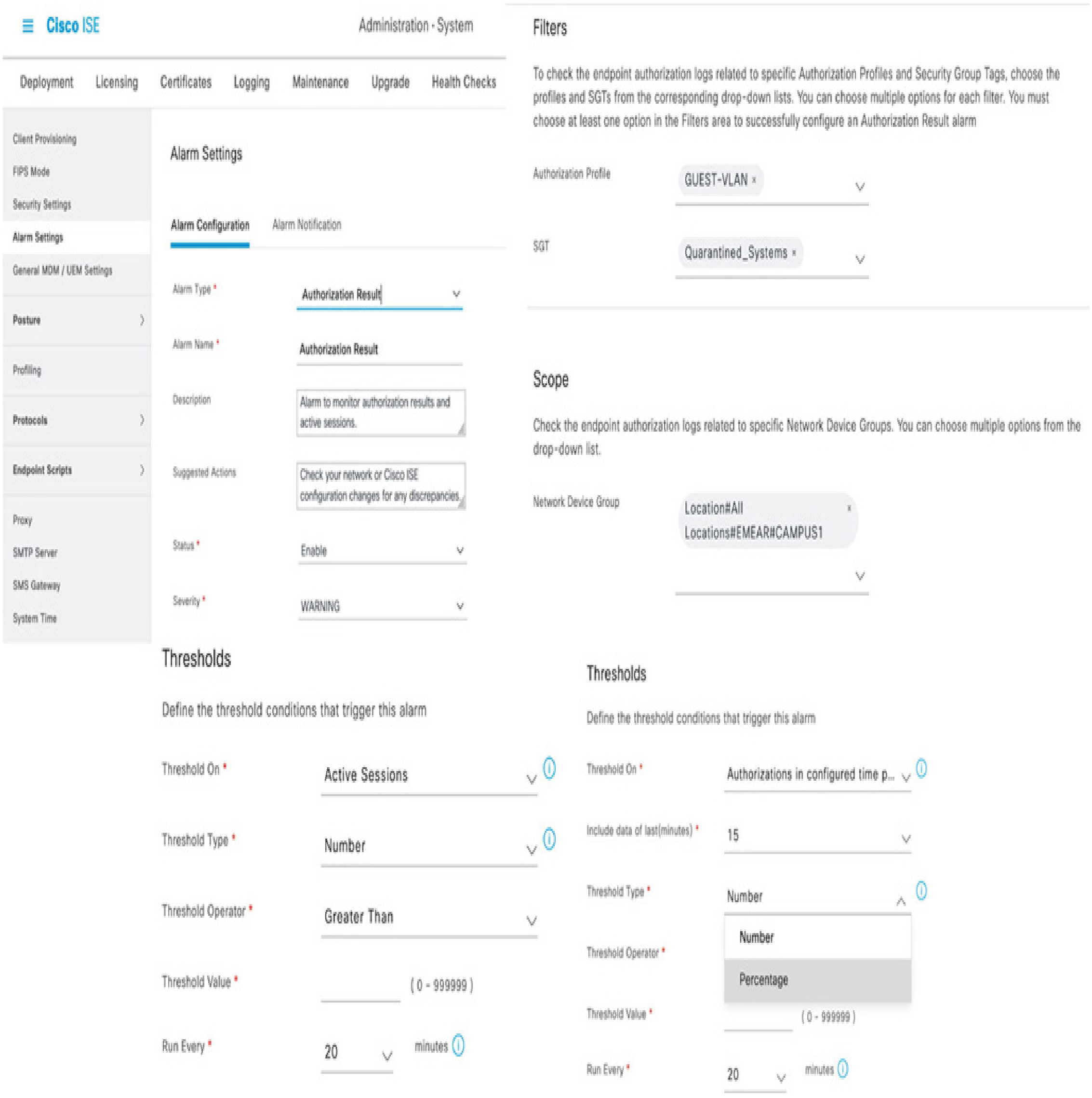
Using these capabilities can lower the number of rogue or unmanaged systems that may exist within the network architecture, allowing a differential access policy to be applied and the principles of zero trust network access to be maintained.

Monitoring Authorization Health

With a centralized server–based authentication and authorization architecture, network access control intelligence shifts away from the edge.

While this approach provides many benefits, it is not without its own challenges. How can a change at the authorization policy level executed by a AAA server or solution operator in Munich, Germany, for a site in Melbourne, Australia, be confirmed to be working properly? How can the appropriate planning take place to ensure that a large-scale outage will not happen after the configuration has been applied? Which measures can be applied to ensure that NAC configurations and policies that are applied are working as desired?

There are many ways to monitor RADIUS authorization activity, from looking at the hit counters in policy rules to verifying report statistics. One of the most useful but very poorly documented (at the time of writing) features available to identity and NAC teams is the use of the RADIUS Authorization Alarms setting in ISE, as shown in Figure 9-23. The GUI output shows a view from the Alarm Settings page, where an Authorization Result ruleset can be written. The figure shows configurable thresholds, locations, and high water marks that can result in an event trigger. This capability allows you to generate threshold-based rulesets based on the number or percentage for hits against authorization rules. You can define these rules per location, matching both the authorization profile and SGTs as filtering criteria.



Alarm Settings

Alarm Type * Authorization Result

Alarm Name * Authorization Result

Description Alarm to monitor authorization results and active sessions.

Suggested Actions Check your network or Cisco ISE configuration changes for any discrepancies.

Status * Enable

Severity * WARNING

Thresholds

Define the threshold conditions that trigger this alarm

Threshold On * Active Sessions

Threshold Type * Number

Threshold Operator * Greater Than

Threshold Value * (0 - 999999)

Run Every * 20 minutes

Filters

To check the endpoint authorization logs related to specific Authorization Profiles and Security Group Tags, choose the profiles and SGTs from the corresponding drop-down lists. You can choose multiple options for each filter. You must choose at least one option in the Filters area to successfully configure an Authorization Result alarm

Authorization Profile GUEST-VLAN ×

SGT Quarantined_Systems ×

Scope

Check the endpoint authorization logs related to specific Network Device Groups. You can choose multiple options from the drop-down list.

Network Device Group Location#All Locations#EMEAR#CAMPUS1 ×

Thresholds

Define the threshold conditions that trigger this alarm

Threshold On * Authorizations in configured time p...

Include data of last(minutes) * 15

Threshold Type * Number

Number

Percentage

Threshold Operator *

Threshold Value * (0 - 999999)

Run Every * 20 minutes

Figure 9-23 *Raising Authorization Alarms Using Cisco ISE*

When configured in the correct way, low watermark or high watermark thresholds can be configured for the respective authorization profile hits per location. As a result, you can monitor what would be considered healthy versus unhealthy to rapidly identify if something in the applied policy may be causing issues. For instance, if the rule in the RADIUS policy is to apply a quarantine system's SGT in scenarios where other policies are not being hit, monitoring the thresholds for this SGT can provide you with a rapid means to see if a misconfiguration may be in place that is sweeping up systems that should normally be hitting other policies. To view this data, you can visualize the export of these alarms via logs by using ISE Log

Analytics, as described earlier, or other third-party systems such as Splunk, Kibana, or Logz.io.

Customer Use Cases

As we have described in this chapter, the use of authentication and authorization in modern networks provides some powerful tools to create a dynamic architecture to quickly adapt and respond to the state of systems in the network, thus ensuring that only the needed access is granted for the tasks and requirements at hand. When applied correctly, these tools can help ensure that a highly scalable and secure architecture can be deployed. These concepts—while often described in the context of enterprises—transcend across business types and technology verticals.

Using Dynamic Policy to Improve Real-World Challenges

Let’s look at a real-world example where authorization rules were applied to fix a problem. In this case, a customer had a library in a small beach town on the northeast coast of Australia. The library, located on a main street that was popular with tourists and backpackers, provided free Internet access to users. While this Internet access attracted interest during the day, the inability to properly geo-fence (limit the Wi-Fi signal) to within the premises meant that the signal reached outside of the building.

Normally, you might think that having a few users utilizing free Internet access outside operating hours is not really a big deal, especially if the bandwidth commissioned is at a flat usage rate and it doesn’t disrupt any other activities from the business. The issue that the library observed, however, touched on the physical world around its location.

In northern New South Wales, the weather is relatively warm most of the year, with balmy summer nights of 28 degrees Celsius. This warm climate often leads to jet-lagged tourists and backpackers spending time out in the streets until the early hours of the morning. Due to high mobile data roaming costs, the library, initially offering free Wi-Fi access 24/7, became a hot spot of activity after hours, leading to backpackers and tourists sitting around the perimeter utilizing the free service.

The challenge with this pattern of usage was that not everybody who decided to sit around the perimeter was conscientious enough to clean up after themselves, often leaving a mess of beer bottles, soft drink cans, and cigarette butts around the entry and the perimeter. While the library staff initially accepted this behavior by putting up signs around the property asking users to be kind enough to clean up after themselves, sadly, not everybody heeded the shared guidance.

In the end, the library decided to explore time-based network authorization as a means to address this problem, permitting guest network access between the hours of 08:00 and 19:00, and then restricting the network access to only corporate systems thereafter. To achieve a reliable and robust solution here, three redundant NTP servers were used to ensure that a valid time synchronization was in place, applied to the wireless controller infrastructure, and ISE. By ensuring that the NTP configuration was applied consistently across the ISE RADIUS server and wireless controller, the library ensured that the correct time enforcement activities could take place, and the web portal hosting the server-side certificates for the guest portal could ensure that a valid and sane clocking source allowed certificate validity to be verified. Since the library applied this change, the challenge they once faced has been resolved.

Expediting System and Workstation Patches

Maintaining system images and patches and security updates can be a challenge, especially when organizations need to deal with a multitude of active systems in their environments. The constant change in hardware, drivers, and relevant system images can sometimes feel like a game of whack-a-mole, which requires significant time and effort spent to ensure that the right levels of validation are performed prior to an image release to avoid unwanted outage scenarios due to incompatibilities that may arise between all of the components.

The next challenge after validation of a viable build and image for the relevant operating system is actually getting users to apply the fix or update in a timely manner. IT workstation teams need to find the right balance between forcing a reboot of a system that might lead to a potential loss of data from open documents and the corporate security needs.

One customer IT team found an innovative approach to expedite such updates in its corporate environments. While previous experience in the organization had shown that the executives, directors, and leadership were not happy with forced reboots unless they could be mapped to a real and tangible security threat that outweighed the business costs of potential data loss on reload, they did want to maintain compliance within environments.

To achieve this goal, the IT team discussed which websites or applications were being used within the company for activities that may not be 100 percent business related. The team put together a list of websites related to sports, cooking, social media, health and fitness, and e-commerce. With this list in hand, the team began to craft a new QoS policy that would provide a rate limit of 128 kbps to such sites, reducing the user experience for “leisure” sites to a very slow speed. Business-relevant content, on the other hand, would be mapped to much higher speeds and protected in the network via QoS policies and mappings that would correspond to SD-WAN IP SLA protection mechanisms to ensure resiliency for anything related to the business.

To avoid upsetting users unnecessarily, the team decided that the right approach to apply this new policy would be through an authorization ruleset that would be dynamically applied on a per user basis. It would be pushed to users only if they decided to defer their software patches and updates past two notification periods. The flow of these events is shown in more detail in Figure 9-24.

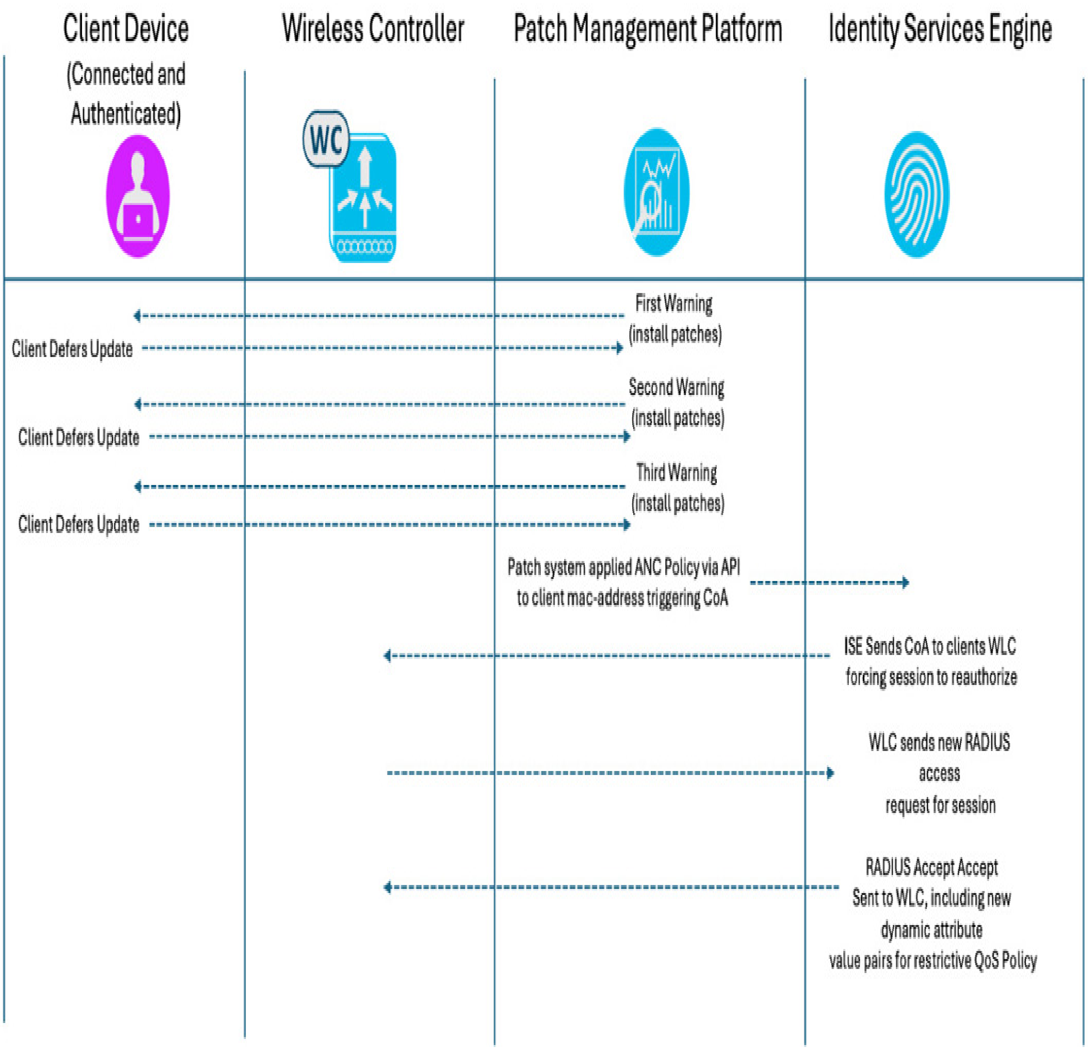


Figure 9-24 *Flow Diagram Illustrating QoS Enforcement Based on Patch Management Status*

After these adjustments were applied, the change in user interaction and speed to update systems was rapid. Prior to taking this approach, the IT team observed that users deferred updates up to 10 times prior to execution. After the change, the number of deferrals went down to an average of 4 times.

Summary

In the challenging and ever-evolving landscape of network security, we've come to recognize that adhering a zero trust security strategy is more than just a strategic move—it's a necessity. The historical approach of fortifying the network perimeter and granting implicit trust to all users within has been outpaced by the demands of today's dynamic digital environments and related threat landscape.

Authentication and authorization are key elements of modern networks and applications. Historical approaches of simply trying to secure the perimeter of a network and implicitly trusting all users within it have proven to not be workable models of operation, lacking the security needed to maintain modern principles of zero trust network access. The new modus operandi has become authenticating and then assigning and dynamically adjusting the levels of authorization for network access based on both ever-changing organizational as well as user needs.

Authentication and authorization stand at the forefront of this evolving landscape, serving as critical components for safeguarding our networks and applications. The zero trust strategy teaches us that trust is a privilege that must be earned and verified continuously, not a default state granted by mere network presence.

Chapter 10. Quantum Security

In this chapter, you will learn the following:

- Basics of quantum computing
- Security adversaries of quantum computers
- Methods to secure against quantum adversaries

What Is Quantum Computing?

You will hear about *quantum computers* in the news and popular press nearly daily. Quantum computing is the new way of computing. These quantum computers are not smaller and more advanced versions of traditional silicon chip-based computers; they represent a new paradigm of computing based on the quantum physics phenomenon. While the focus of this book is on security, because quantum computing is a new technology, it has yet to be known to IT experts to develop skills relevant to future security needs and remain competitive in the market. We have decided to cover the concepts of quantum computing and computers in detail to form a solid foundation for the topic before you start looking at the security implications of quantum technology. It would be best to approach this chapter with a fresh mindset because quantum physics is unfamiliar to most of us in the IT industry. Some of the concepts presented in this chapter might feel hard to relate to or understand, but those are based on proven quantum mechanics theories and experiments.

As we start delving into the details of quantum computing, we need to understand computing and how humans have evolved their computing abilities. The process of calculating or determining something using mathematical or logical methods is known as *computing*. The human brain is one of the finest computing devices; we can put it under the category of biological computers. Throughout history, humans have used different tools for calculation. Early humans used even simple tools like stones or sticks for primitive forms of counting and basic arithmetic, which later advanced into sundials and stone calendars, which were precursors to computing. Historically, humans have used various mechanical devices for calculations, such as the Turing machine and abacus. Modern computing is based on electronic machines. The current generation of computers uses integrated circuits (transistor-based) circuitry for information processing.

For the discussion in this chapter, we will call current-generation chip-based computers *classical computers*, regardless of how fast or advanced they are. Classical computers are based on the binary numbers 0 and 1. Some of the early classical computers, such as ENIAC, were based on vacuum tubes and were huge, occupying an entire floor of a building. The next generation of computers started using transistors instead of vacuum tubes. By the 1970s, integrated circuits led to the development of personal computers. Over the years, we have made significant progress with classical computers; current-generation computers are extremely fast, efficient, and able to multitask. We have also moved from the traditional form factor of computers (desktop and laptop) to new devices such as phones and tablets.

The Need for New Computing Technologies

You must be wondering why we need a different computing method. We are happy with what our classical computers can do and focus on making them smaller, more efficient, and faster. As regular home users, we don't use computers for many complex tasks such as someone in a pharmaceutical company or scientists in astrophysics or quantum mechanics would do to simulate the structure of a virus, study the interaction of quantum particles, or simulate nuclear particle collisions. It is assumed that the availability of fully functional quantum computers could have further accelerated the development of COVID-19 vaccination by many months. Many other fields, like material science and chemistry, require complex simulation work at higher speeds. As you will learn in later sections, many of these computations require probabilistic calculations. Further combining artificial intelligence with quantum computing creates a new paradigm, enabling unparalleled data processing and decision-making at quantum speeds. This synergy transforms problem-solving, making the impossible achievable in science, security, and beyond.

While the current generation of computers is fast, they must be quicker for scientific computations. The first idea of using different techniques for computing came from theoretical physicist Richard Feynman. He recognized that classical computers struggle to simulate quantum systems efficiently because they can think of any state as either 0 or 1. It is difficult to simulate the concepts of quantum physics like entanglement and superposition (explained in detail later in this chapter) using classical computing methods. This led to the creation of a computing device based on the quantum physics phenomenon that can be in multiple states simultaneously. This concept later developed into a quantum computer. These quantum technology–based computers excel in tasks like cryptography, material science, and large-scale optimization, solving problems beyond the reach of classical computers. By pursuing quantum technology, we unlock capabilities that can revolutionize industries, drive scientific breakthroughs, and address complex global challenges efficiently and securely.

How Quantum Computing Is Different

Classical computers use the concept of a *bit*. A bit represents one of the two distinct logical states, 0 or 1. In transistor-based computers, state 0 corresponds to the off state, which means no voltage is applied to the transistor gate, while state 1 means that voltage is applied to the gate and current flows through it. As these are discrete states, their measurement is easy and with high certainty. Everything in classical computers is represented by a number and is presented in binary notation. Audio, video, text—everything is in numbers, which can be organized in bits, nibbles (4 bits), and bytes (8 bits); however, a programmer or user doesn't have to deal with these bits and bytes because our programming languages abstract all these details from us, and we code using programming syntax. You will notice quantum computers are different in how they store information.

The fundamental elements of the quantum computer are called *quantum bits*, or *qubits* in short form. These are quantum mechanical objects and can be presented by photons, ions, and atoms. Based on the material used for creating a qubit, it will be an atomic qubit, photonic qubit, or ionic qubit. Other materials can also be used for creating qubits. As there are quantum mechanical objects, they can be prepared to represent the quantum superposition state, where a qubit will represent aspects of both 0 and 1 at the same time. This means that apart from discrete values 0 and 1, a qubit can take an infinite number of positions between 0 and 1. These states can be visualized using the Bloch sphere, as shown in Figure 10-1.

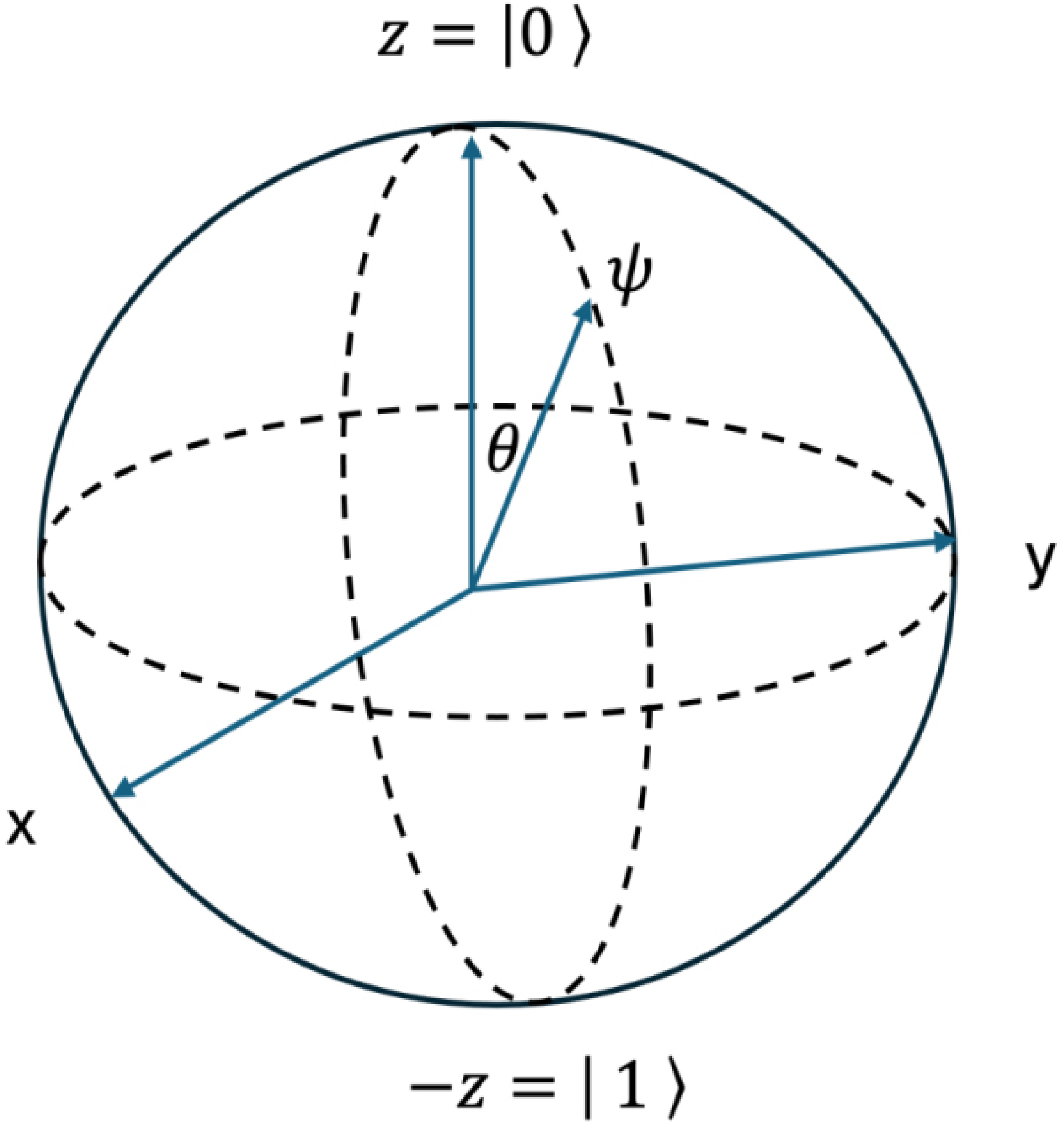


Figure 10-1 *Bloch Sphere*

The z-coordinate represents the probability of measuring the qubit in the |0⟩ or |1⟩ state.

- When the Bloch vector points to the north pole ($z = 1$), the qubit is in the state |0⟩ (definite).
- When the Bloch vector points to the south pole ($z = -1$), the qubit is in the state |1⟩ (definite).
- Points between the poles represent a superposition.

  The x and y coordinates represent the phase and relative amplitude of the qubit's superposition. The x-y plane gives the phase (complexity) of the superposition.
- A point on the equator (x-y plane) means the qubit has an equal superposition of |0⟩ and |1⟩.

Quantum Superposition

Superposition is the ability of the quantum system to be in multiple states at the same time. This state is retained till it is measured. This means that a qubit can be in any state, which is some portion of 0 and 1 at a given time, but when a quantum system is observed or measured, it falls to one of the discrete states 0 or 1. Now, this may be hard to grasp if you are not familiar with quantum physics concepts, but this is what gives power to the quantum computers.

Let’s consider an example where a qubit is made of atoms, and we use electron spin for measurement. In this case, apart from discrete states of 0 and 1, when put in a superpositioned state, a qubit will represent some portion of both 0 and 1at the same time as represented in Bra-ket notation (a mathematical notation to represent the state of a qubit), as shown in Figure 10-2. Quantum states can also be represented in vector notation. This is very helpful when gate operations are performed on the qubits.

Electron spin-based qubit - Intrinsic angular momentum

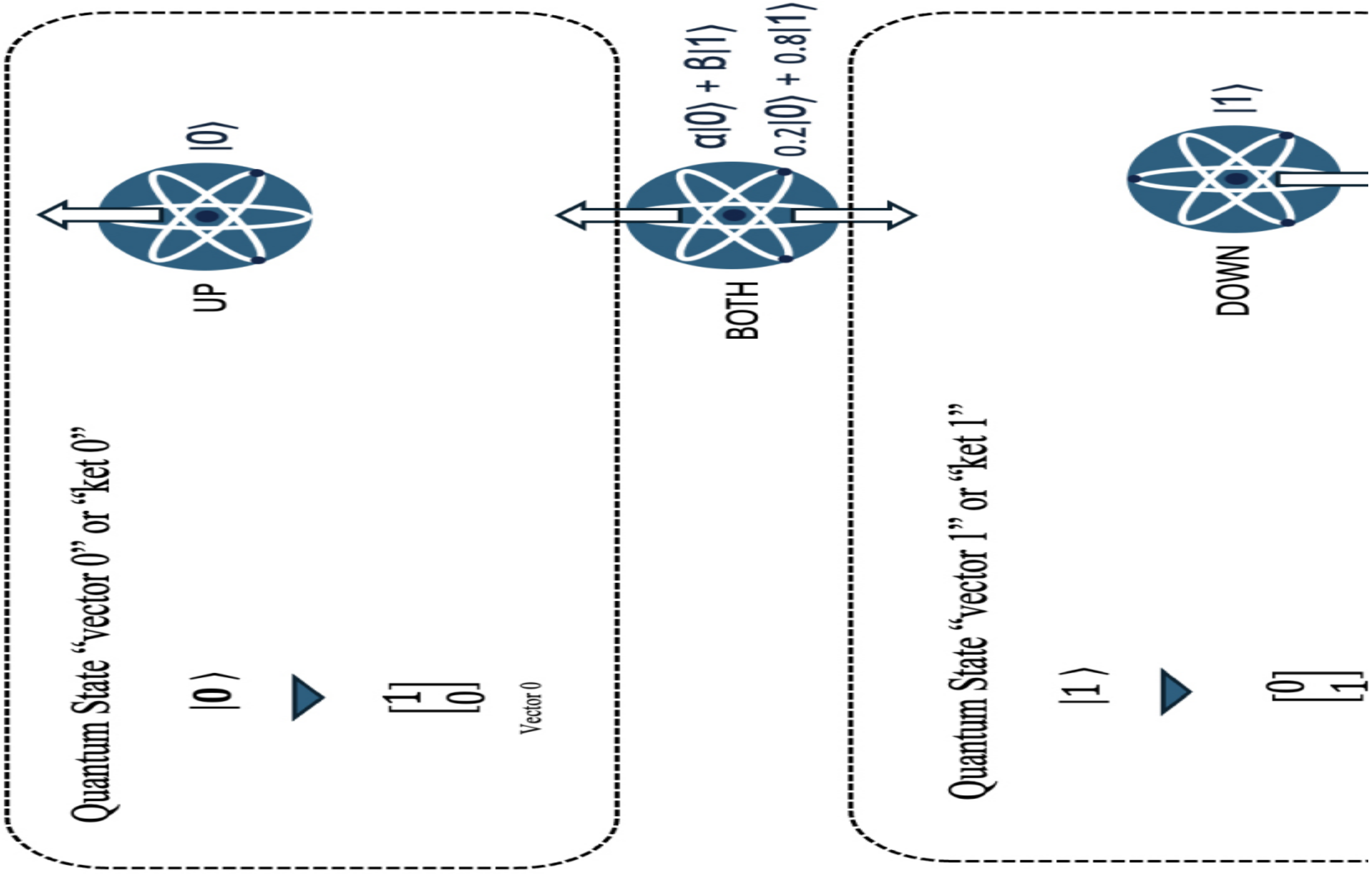

Quantum State “vector 0” or “ket 0”
|0⟩
[1 0]
Vector 0
UP
|0⟩
BOTH
α|0⟩ + Β|1⟩
0.2|0⟩ + 0.8|1⟩
Quantum State “vector 1” or “ket 1”
|1⟩
[0 1]
DOWN
|1⟩

Figure 10-2 *Bra-ket Notation for the Qubit*

$$|\psi\rangle = \alpha|0\rangle + \beta\,|1\rangle$$

In this equation, alpha (α) and beta (β) are the probability amplitudes of the values 0 and 1.

This means

$$|\alpha|^2 = Probability\ of\ measuring\ 0$$

$$|\beta|^2 = Probability\ of\ measuring\ 1$$

Quantum superposition allows a quantum computer's qubit to exist in multiple states (|0⟩ and |1⟩) simultaneously, enabling parallel computation. This concept of superposition is compelling. Let’s assume you have two classical bits and two quantum bits. How much information can be represented by this set? With classical bits, each bit can have either 0 or 1. This means we can have the four combinations 00, 01, 10, and 11, which means 2^*n* states, where *n* is several bits. As shown in Figure 10-3, in quantum computing, each qubit can have both states at the same time; this means that a single state of a qubit set can represent all the four classical combinations.

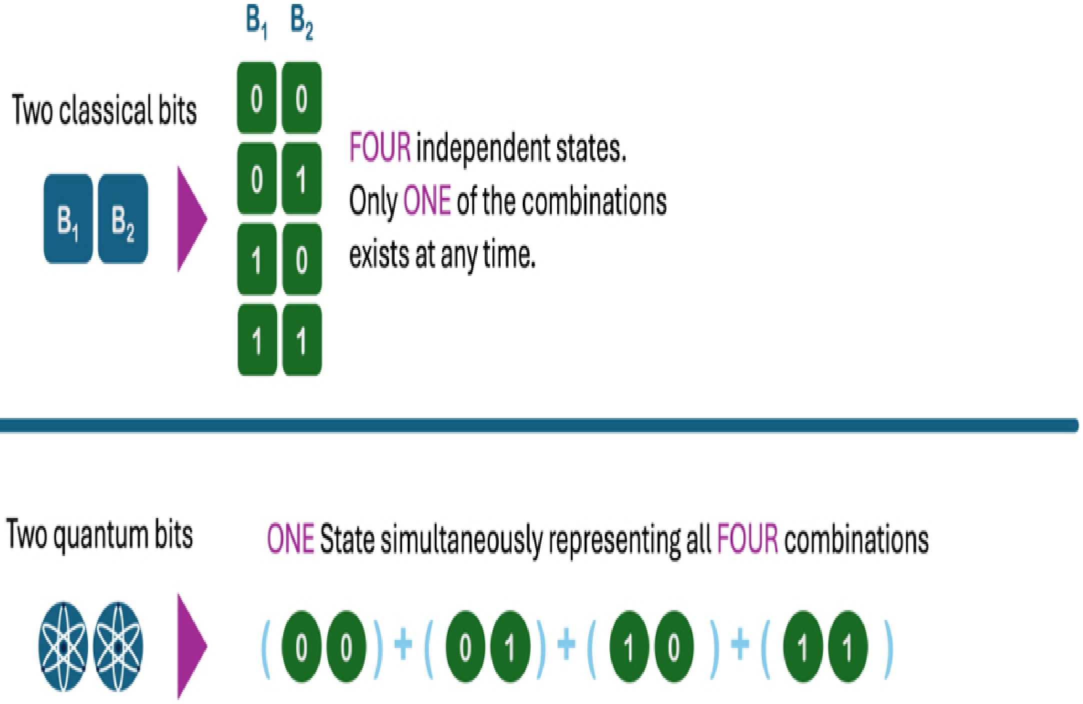
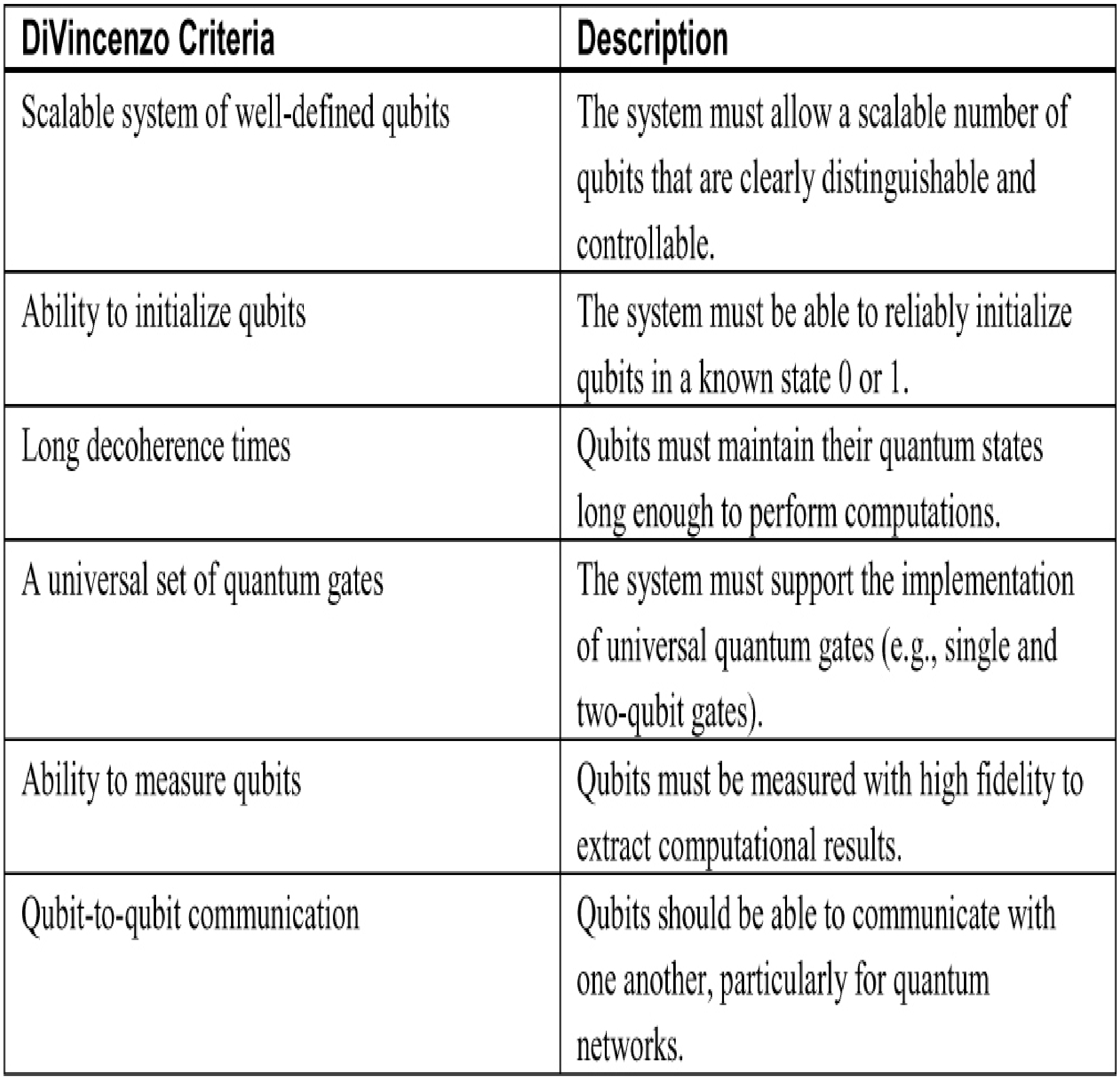


Figure 10-3 *Classical vs. Quantum States Representation*

Qubit Modalities

Several quantum systems might work as qubits. Examples include the electronic states of an ion, the electron spin of a phosphorus atom in silicon, the nuclear spin of a defect in a diamond, and even man-made “artificial atoms,” which are electrical circuits that behave like quantum systems. There are basic qualities should these systems be considered as candidates for quantum computing. The DiVincenzo criteria outline essential requirements for a physical system to function as a quantum computer. These criteria ensure that qubits can be effectively manipulated and measured for quantum computation.

Table 10.1 *DiVincenzo Criteria for Functional Quantum Computer*



DiVincenzo Criteria	Description
Scalable system of well-defined qubits	The system must allow a scalable number of qubits that are clearly distinguishable and controllable.
Ability to initialize qubits	The system must be able to reliably initialize qubits in a known state 0 or 1.
Long decoherence times	Qubits must maintain their quantum states long enough to perform computations.
A universal set of quantum gates	The system must support the implementation of universal quantum gates (e.g., single and two-qubit gates).
Ability to measure qubits	Qubits must be measured with high fidelity to extract computational results.
Qubit-to-qubit communication	Qubits should be able to communicate with one another, particularly for quantum networks.

One of the challenges with qubits is reducing quantum decoherence time. *Quantum decoherence time* refers to a time when a quantum system can maintain its quantum state before interacting with the environment, causing it to lose its quantum properties. In simpler terms, it's the period during which a quantum particle, like a qubit, can stay in a delicate superposition state (where it can exist in multiple states simultaneously). Once decoherence happens, the system behaves more like a classical system, losing the advantages that make quantum computing possible. Ongoing research is happening in this field at the time of writing this chapter.

Quantum Entanglement

If you are a science fiction movie fan, you may have heard the term *quantum entanglement* often. It is a fascinating phenomenon in quantum physics where two or more particles become linked to each other in such a way that one particle directly influences the state of the others, regardless of the distance between them. This means that any change in the property of one particle, like spin or polarization, will instantly affect its entangled partner. This works even if the particles are in different cities, countries, planets, or galaxies. This is strange behavior with no explanation. Einstein used to call this "spooky action at a distance."

Quantum computers use this phenomenon for calculation and do entanglement between a set of qubits. Qubit entanglement can be of two types—same or opposite. Using the same entanglement, one qubit behaves exactly the same as the first one. In opposite entanglement, a qubit behaves opposite of its peer. Once qubits are entangled, states can be predicted in advance. Let's say we have two entangled qubits—black and white. Figure 10-4 shows the states when put in opposite entanglement.

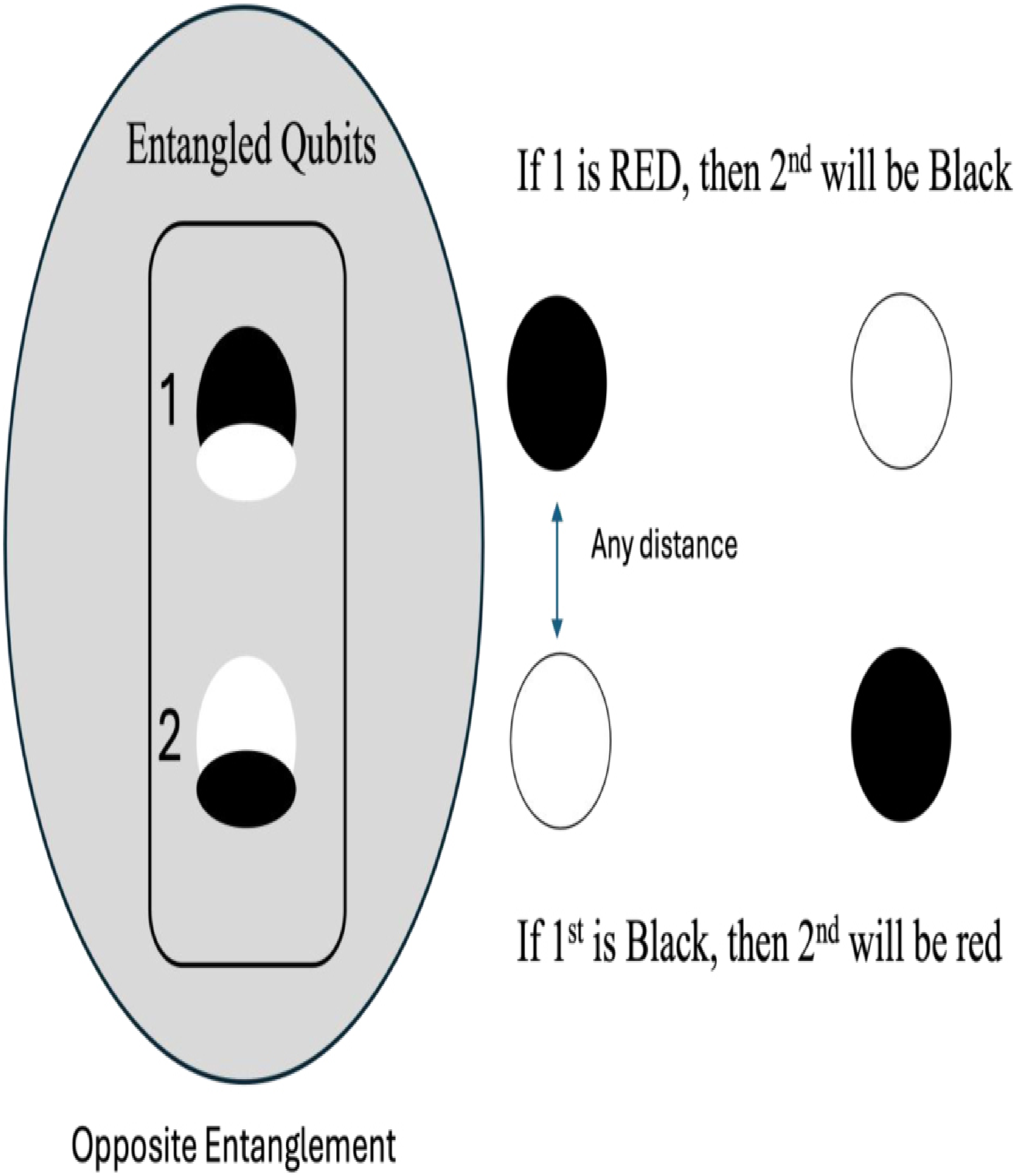

Entangled Qubits
If 1 is RED, then 2nd will be Black
1
Any distance
2
If 1st is Black, then 2nd will be red
Opposite Entanglement

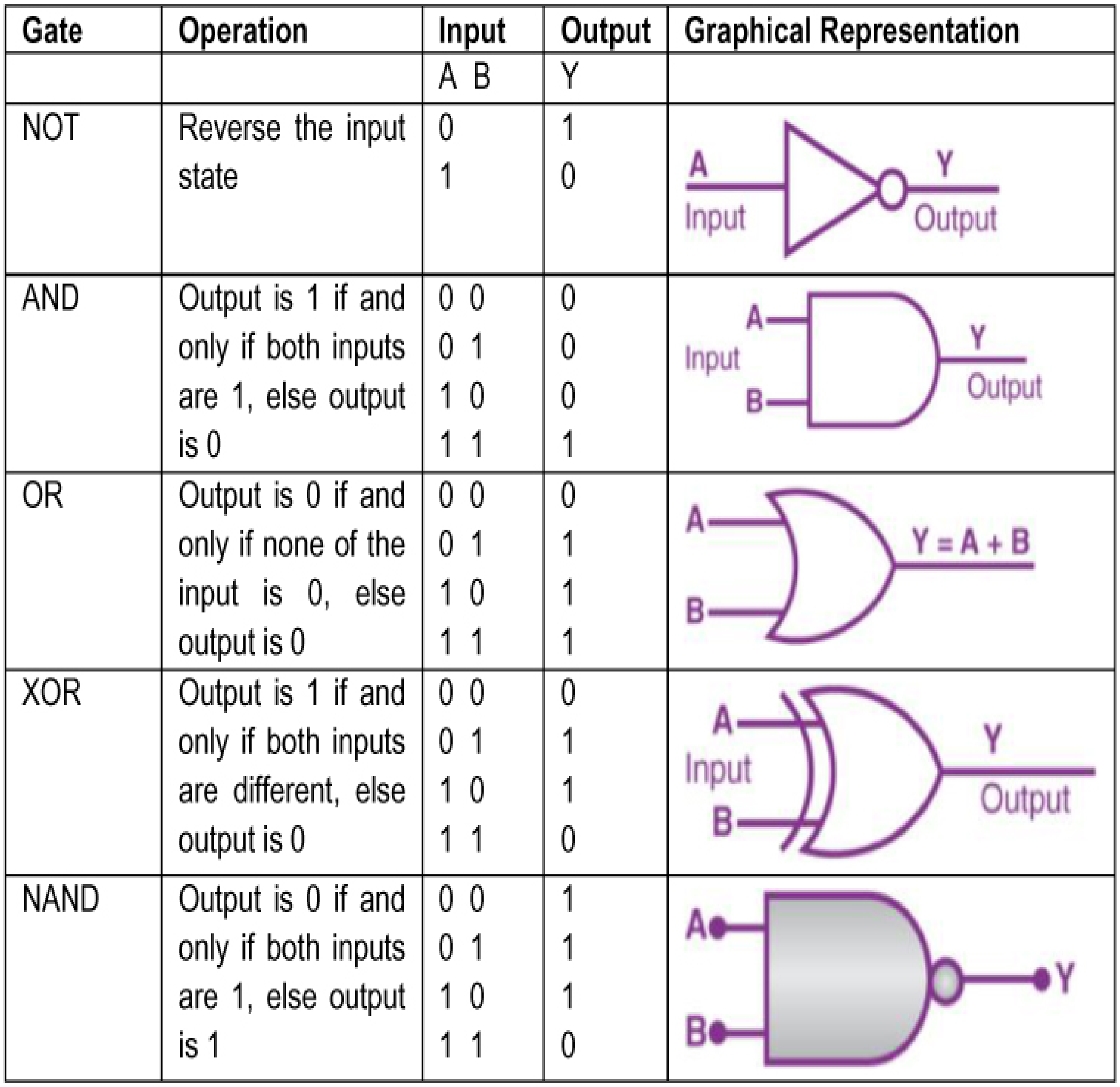
Figure 10-4 *Quantum Entanglement*

Qubits are put in superposition or entanglement states using quantum gates, which you will learn about later in this chapter.

Gate Operations

Logic gate operations form the base of the information processing in classical gates. These are the fundamental components of processing binary information. The classical gates manipulate bits in deterministic ways by following the rules of Boolean algebra. We then use the combination of these gates to build more complex circuits for processing tasks like addition and subtraction. Let’s look at some of the common classical gate operations in Table 10-2. This table might take you down memory lane to the early days of your computer classes in school.

Table 10.2 *Gate Operations*



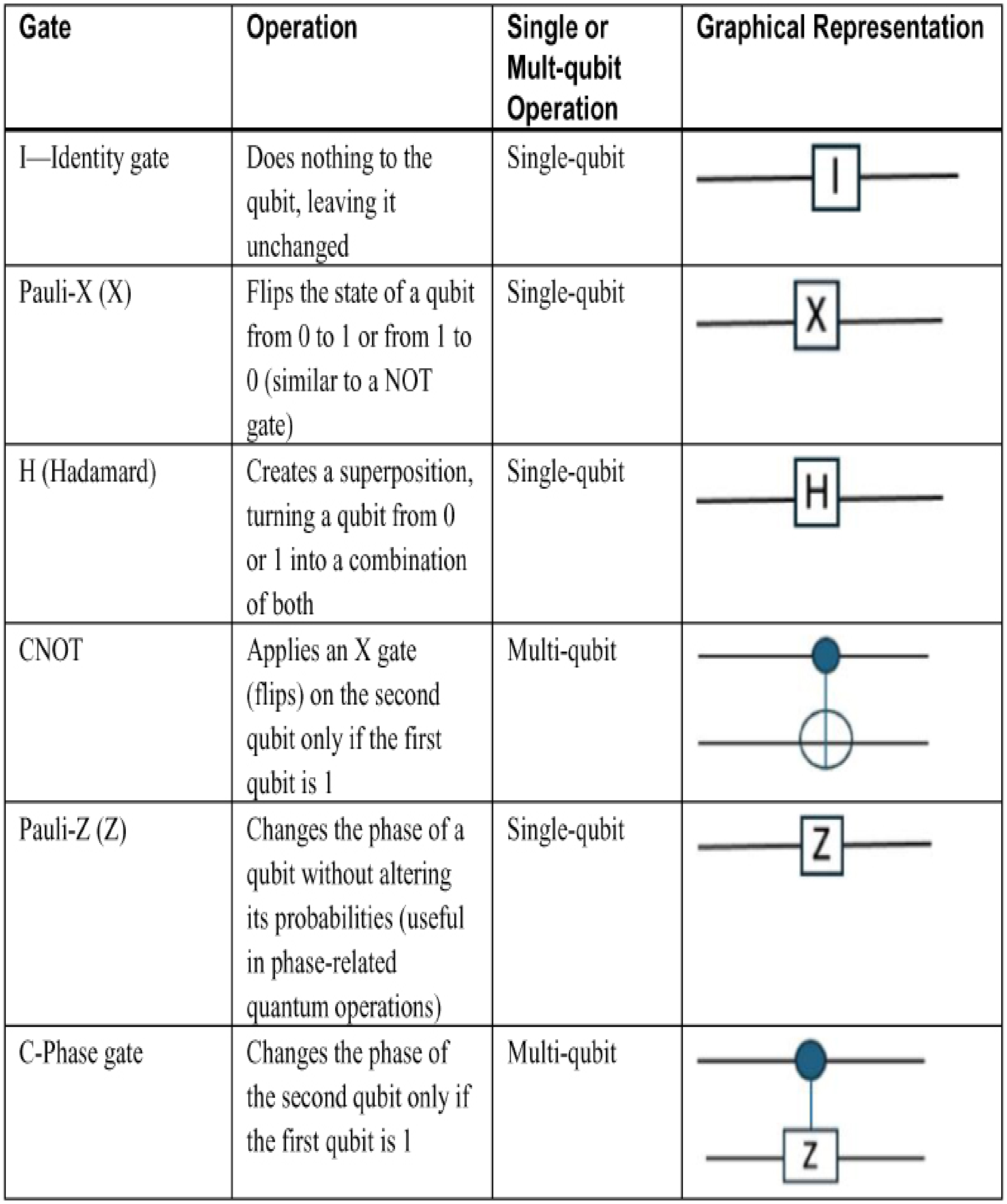
Gate	Operation	Input	Output	Graphical Representation
		A B	Y	
NOT	Reverse the input state	0 1	1 0	
AND	Output is 1 if and only if both inputs are 1, else output is 0	0 0 0 1 1 0 1 1	0 0 0 1	
OR	Output is 0 if and only if none of the input is 0, else output is 0	0 0 0 1 1 0 1 1	0 1 1 1	
XOR	Output is 1 if and only if both inputs are different, else output is 0	0 0 0 1 1 0 1 1	0 1 1 0	
NAND	Output is 0 if and only if both inputs are 1, else output is 1	0 0 0 1 1 0 1 1	1 1 1 0	

Similar to classical logic gates, quantum gates operate on qubits. Remember, qubits are unique because they can exist in a superposition of 0 and 1 simultaneously and can be entangled with other qubits. Quantum logic gates perform operations that transform qubit states but with two key differences when compared with classical gates:

- **Reversibility:** All quantum gates are reversible, meaning you can "undo" an operation, which is not always true for classical gates. For instance, in classical logic, the AND gate is not reversible because knowing the output doesn't tell you what the inputs were. Whereas if you put two quantum Controlled-NOT (CNOT) gates in series, they will nullify the state change.
- **Quantum Superposition and Entanglement:** Quantum gates leverage superposition and entanglement, two unique quantum properties, allowing them to perform highly complex calculations that are impossible or extremely inefficient for classical computers.

Table 10-3 summarizes some of the common quantum gates. Some of these gates operate on a single qubit, whereas others operate on multiple. From a network security perspective, you don’t have to remember any of this, but it is important to understand the fundamentals of quantum computing to explain how and why quantum security needs to be considered as part of the security strategy for any organization.

Table 10.3 *Quantum Gates*



Gate	Operation	Single or Mult-qubit Operation	Graphical Representation
I—Identity gate	Does nothing to the qubit, leaving it unchanged	Single-qubit	I
Pauli-X (X)	Flips the state of a qubit from 0 to 1 or from 1 to 0 (similar to a NOT gate)	Single-qubit	X
H (Hadamard)	Creates a superposition, turning a qubit from 0 or 1 into a combination of both	Single-qubit	H
CNOT	Applies an X gate (flips) on the second qubit only if the first qubit is 1	Multi-qubit	
Pauli-Z (Z)	Changes the phase of a qubit without altering its probabilities (useful in phase-related quantum operations)	Single-qubit	Z
C-Phase gate	Changes the phase of the second qubit only if the first qubit is 1	Multi-qubit	Z

Some of the gate operations are worth discussing in detail because they are used in most quantum circuits or algorithms.

H Gate

The Hadamard gate, commonly known as the H gate, is one of the most fundamental single-qubit gates in quantum computing. Its primary function is to create superposition, a core concept in quantum mechanics. It transforms a qubit from a definite state (either $|0\rangle$ or $|1\rangle$) into a superposition, where the qubit has a probability of being in both $|0\rangle$ and $|1\rangle$ simultaneously.

If the input qubit is $|0\rangle$, the Hadamard gate transforms it into an equal superposition of $|0\rangle$ and $|1\rangle$:

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(\,|0\rangle + |1\rangle\,)$$

In the preceding equation, after applying the H gate, the probability of measuring 0 or 1 is now 50 percent.

Similarly, if the input qubit is $|1\rangle$, the Hadamard gate transforms it into an equal superposition of $|0\rangle$ and $|1\rangle$, but with the state $|1\rangle$ having a negative sign. That is just a representation; it also puts the qubit in a superposition state with the probability of measuring 0 and 1 as 50-50.

$$H|1\rangle = \frac{1}{\sqrt{2}}\ (\,|0\rangle \times\ |1\rangle\,)$$

In Figure 10-5, let’s assume you have two balls, black and white, passing via the H gate. The probability of getting a white or black ball as output is 50-50. The -ve sign indicates an additional aspect of the state. The H gate creates a superposition state. This is the simplest superposition.

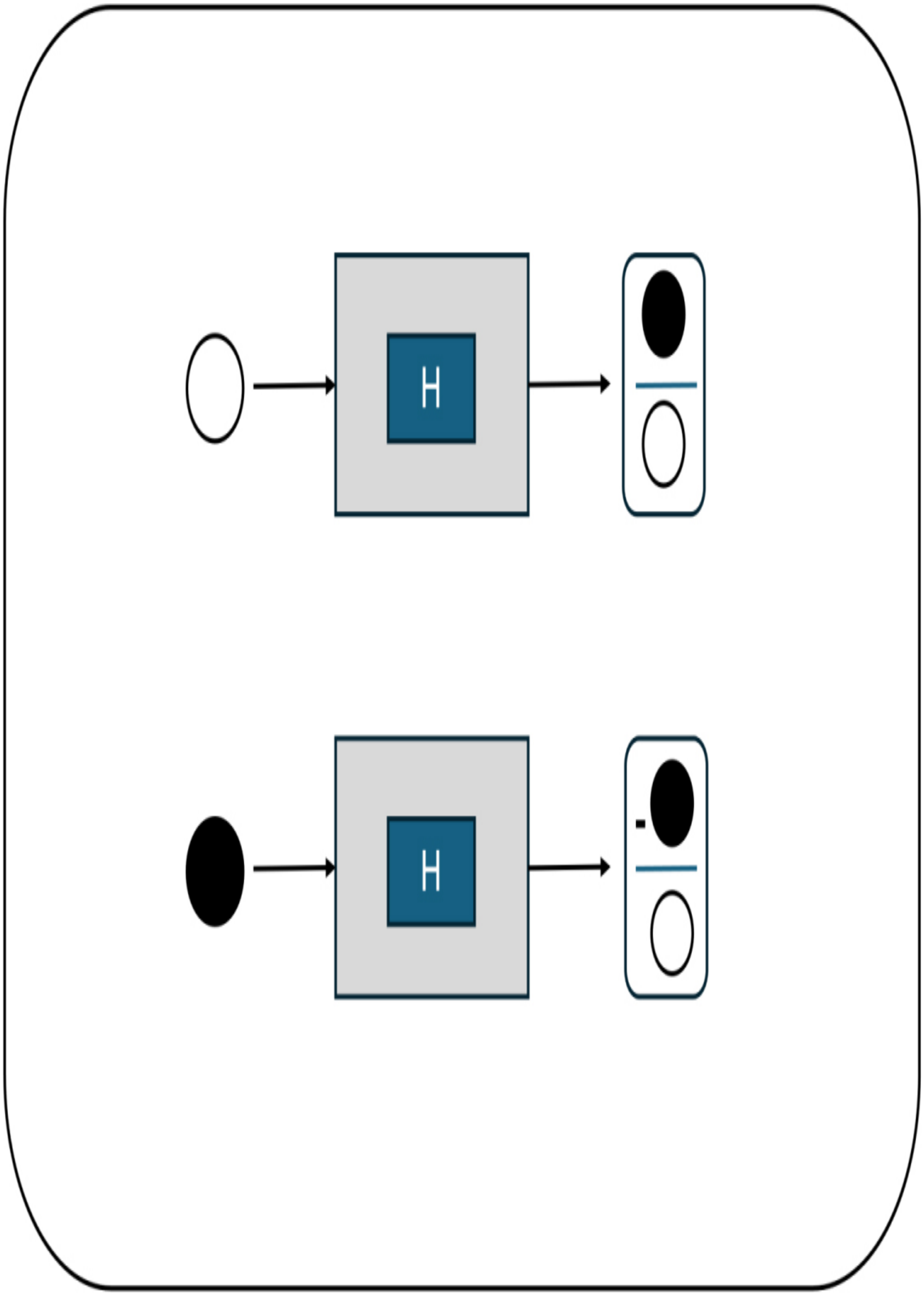


Figure 10-5 *H Gate Operation*

CNOT Gate

The CNOT gate, short for Controlled-NOT gate, is a fundamental two-qubit quantum gate. It is a conditional gate comprising two qubits: a *control qubit* and a *target qubit*. It plays a critical role in quantum algorithms and is central to creating entanglement between qubits. The CNOT gate flips the state of one qubit (called the target qubit) based on the state of another qubit (called the control qubit).

- If the control qubit is in the state $|0\rangle$, the target qubit remains unchanged.
- If the control qubit is in the state $|1\rangle$, the target qubit is flipped (from $|0\rangle$ to $|1\rangle$ or from $|1\rangle$ to $|0\rangle$).

Let's look at an example of this gate operation. Let's assume you have two balls (black represents 1 and white represents 0) passing via CNOT gates, as shown in Figure 10-6. Based on the control qubit, the value state of the target will change. You will notice that the color of the ball changes only when the control bit is in the ON state (in our case represented by the black color).

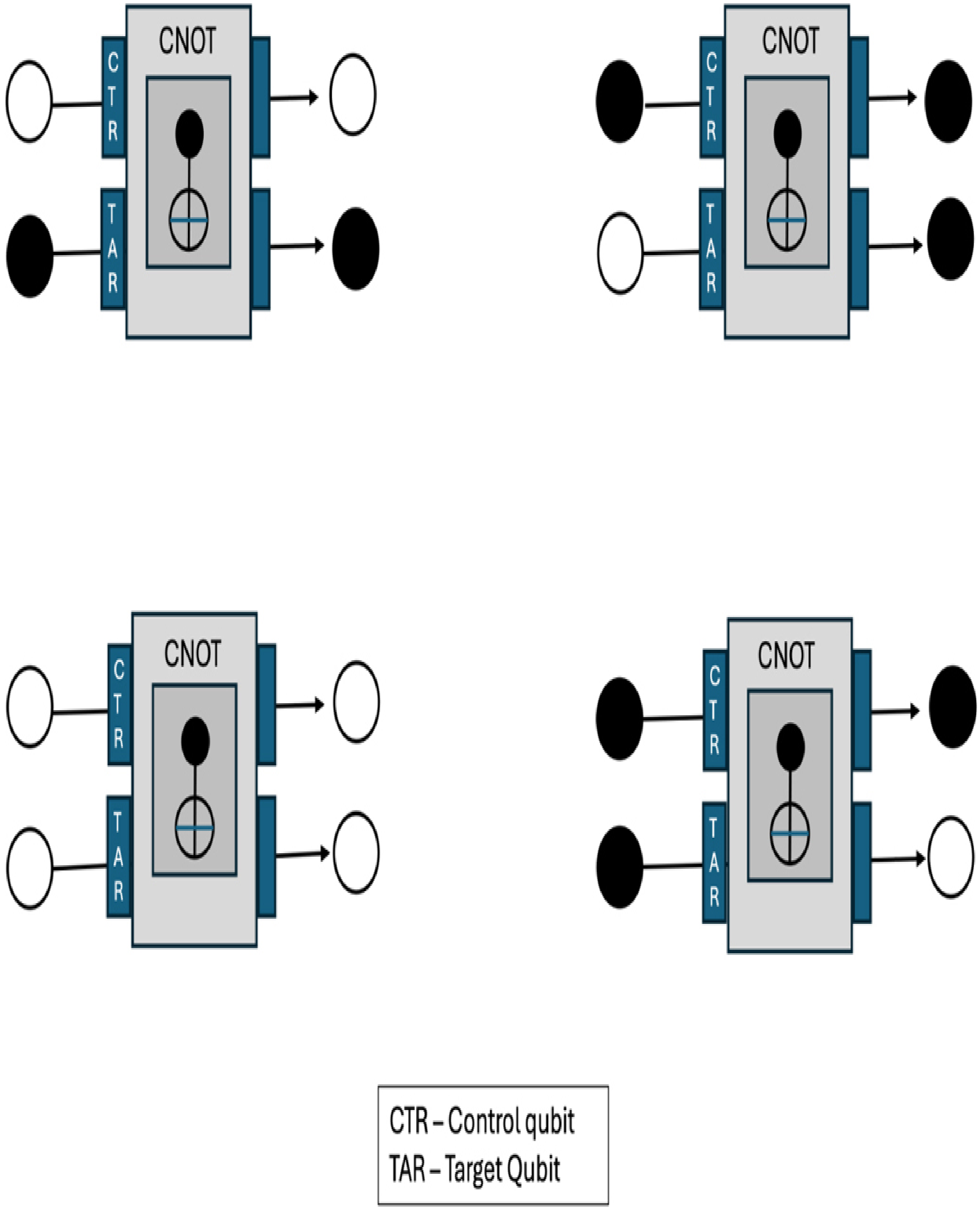


Figure 10-6 *CNOT Gate Operations*

Along with the Hadamard gate (or H gate), the CNOT gate forms part of the set of universal quantum gates. Any quantum operation can be constructed using combinations of these two gates.

Using the combination of H gate and CNOT gate, we can put a pair of qubits in a *Bell state*. This is a specific type of quantum entangled state involving two qubits. In this state, the qubits are perfectly correlated, meaning the measurement of one qubit will directly influence the state of the other, regardless of the distance between them. Bell states represent the simplest and most well-known examples of entanglement, often used in quantum information theory, quantum teleportation, and quantum cryptography.

To create a Bell state from two unentangled qubits, you use a combination of H gate and CNOT quantum gates.

The Hadamard gate is applied to the first qubit to put it into a superposition of |0⟩ and |1⟩. After this operation, the qubit exists in the state:

$$\mathrm{H}|0\rangle = \frac{1}{\sqrt{2}}(\,|0\rangle + |1\rangle\,)$$

This puts the first qubit into a superposition, making it equally likely to be measured as 0 or 1.

Next, the CNOT gate is applied with the first qubit as the control and the second qubit as the target. The CNOT gate flips the state of the second qubit only if the first qubit is in the state |1⟩. The operation entangles the two qubits.

After the CNOT gate is applied, the qubits are in one of the Bell states. If the second qubit starts in the |0⟩ state, this will result in the Φ^+ Bell state, as shown in Figure 10-7.

$$\frac{1}{\sqrt{2}}\,(|00\rangle) + |11\rangle)$$

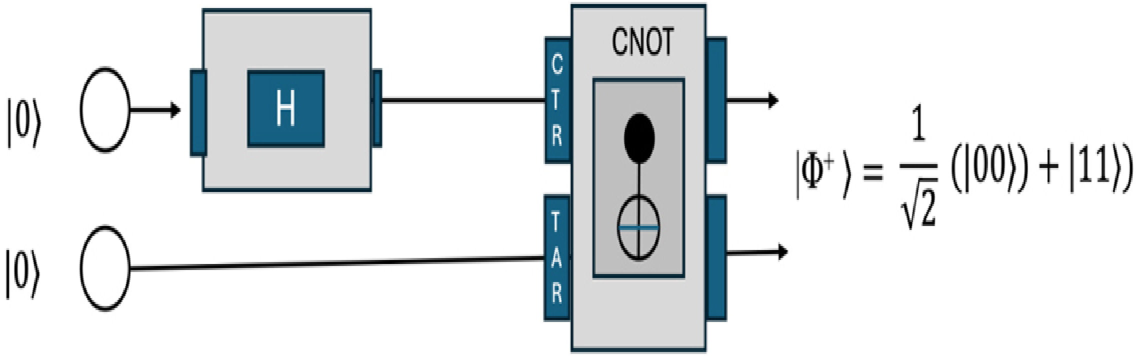

|0⟩
|0⟩
H
CTR
TAR
CNOT
$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Figure 10-7 *Bell State*

Using many such gates, we can create different circuits to process information; one example is shown in Figure 10-8. As explained in the next section, some of them have shown the creation of security adversaries for computer networks.

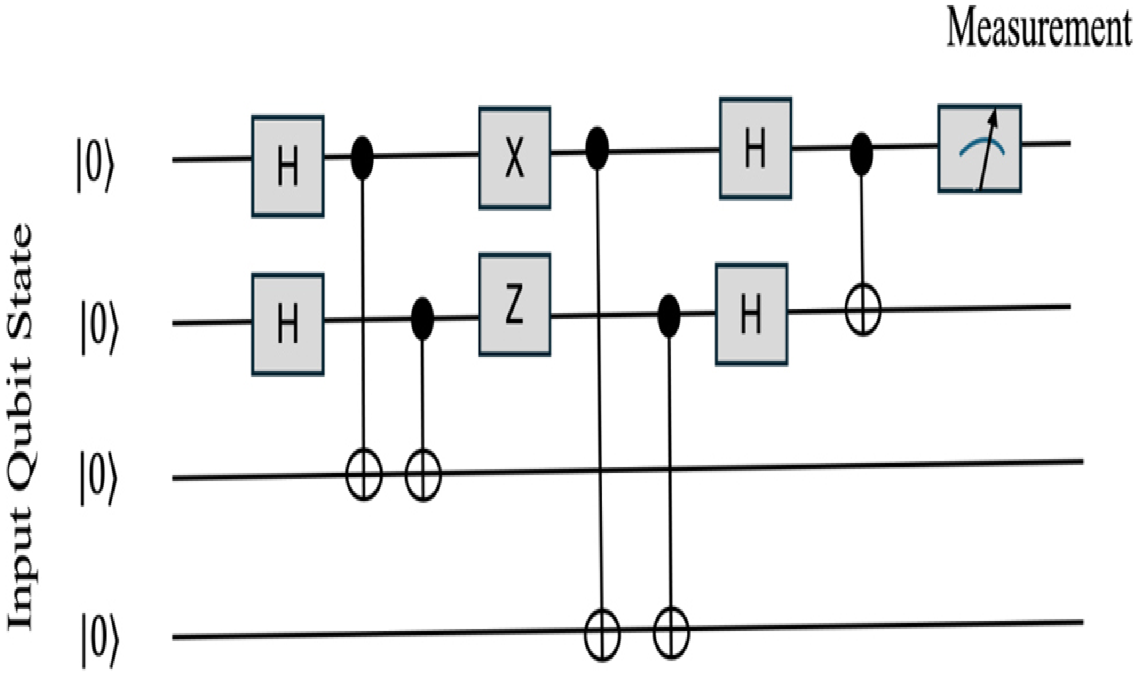


Figure 10-8 *Sample Quantum Circuit*

Current State

At the time of writing this chapter, while we are making great progress toward a fully functional quantum computer with enough qubits, they are still in the infancy stage. They occupy an entire laboratory space that includes the core of the quantum computer and a variety of machines and tools to operate it. You will notice that quantum computers are usually housed inside a huge cylindrical structure. It is necessary to shield the quantum computer from the sources of electromagnetic interference, heat, and so on, because they impact the performance of qubits, as you learned in the previous section. Typically, the core of quantum computers is operated at temperatures less than 20 milliKelvins (this is much colder compared to outer space). Different systems, such as laser beams or microwaves, are used to alter the state of a qubit. Laser-based cooling is common for trapped ion-based qubits. You will always see that classical computers are used to control the inputs toward the quantum computers.

Following are challenges for developing quantum computers:

- **Qubit Stability:** Qubits are very sensitive to noise and heat, which causes them to lose their quantum state (decoherence) quickly.
- **Scaling Up:** It is difficult to manage and control large numbers of qubits and connect them for useful computations.
- **Temperature Requirements:** Quantum computers must be kept at extremely cold temperatures, close to absolute zero, which is costly and complex.
- **Error Correction:** Even small mistakes can ruin results, and fixing errors requires many extra qubits, making the process harder.
- **Quantum Software:** The software and algorithms needed to make quantum computers useful are still in the early stages of development.

Quantum Computing and Emerging Security Threats

Quantum computing is a complex topic, and explaining it in a short chapter is impossible. We hope you got some basic ideas about quantum computers and their operations from the preceding sections of this chapter. In this section, we will pivot back to the book's main theme about security.

Quantum computers can complete probabilistic work in a fraction of the time compared to classical computers. This means that once commercially available and ready to use in general, they will be capable of solving complex mathematical equations. Now that is a problem, because most of the cryptographic algorithms are based on mathematical formulae with the assumption that there is enough complexity that classical computers should not be able to decode them within a couple of centuries. Today, many systems use encryption to protect data. Encryption changes data into a code so nobody can read it unless they have the right key. The most common encryption we use today depends on the fact that normal computers take a long time to factor big numbers. This makes it safe because the process to break the encryption is too slow for the existing classical computers.

With quantum computers, it's different. They can factor in these big numbers very quickly. If a hacker has a quantum computer, they could break the encryption and steal information that is supposed to be secret, like bank details, personal data, or government secrets. This is a big risk for security because the current encryption methods will not work anymore to safeguard against quantum adversaries. A hacker can break current generation encryption if they have access to a quantum computer with enough qubits. They can even store the encrypted communication today and access it later once they have access to a quantum computer. Scientists are working on something called *post-quantum cryptography*. This is a new way of making encryption that quantum computers cannot break. We must find these new methods before quantum computers become powerful and widely available.

Now, let's dig into the details of this. There are two main types of cryptography in use today: symmetric and asymmetric.

Symmetric Keys: In this type of encryption, the same key is used for both encryption and decryption. Both the sender and the recipient must have access to the same key, which can be challenging to distribute securely. German forces used enigma encoding during World War II. An electro-mechanical rotor machine was used to encode plain text into the encrypted text by replacing characters in plain text pseudo-randomly with a different letter according to the wiring of the machine. This enigma machine was an example of one of the first symmetric-based schemes. In 1975, IBM first introduced the Data Encryption Standard (DES), which generates a 56-bit symmetric encryption key using a complex mathematical method. DES was later replaced by the Advanced Encryption Standard (AES), a more secure symmetric key cryptography scheme. But still the main question remained unanswered: How can two parties exchange a symmetric key before they have a secure channel on which to send it?

Some of the challenges with symmetric keys include

- Key distribution is a big challenge because both sender and receiver must have the same key. You require a secure channel to share the encryption keys.
- Because each pair requires separate keys, sharing such key pairs in a large network becomes a scalability issue. For N users, you will require N*(N − 1)/2 keys.
- Because both parties use the same key to encrypt the message, there is no way to prove who encrypted the message. This becomes a problem in legal communication because the sender could deny having sent the message. This means nonrepudiation is not guaranteed while using symmetric keys.
- If the key is compromised, an attacker can decrypt all past and future communication until the key is changed. Therefore, complex key management functions are needed.

The main challenge of symmetric encryption is sharing keys via a secure channel. Later in this chapter, you will learn that despite all these issues, symmetric keys are secure against quantum computers.

Asymmetric Encryption: In this type of encryption, two separate encryption and decryption keys are used. Asymmetric keys comprise a public key (used to encrypt messages) and a mathematically related but unique private key (used to decrypt messages). The public key encrypts the data, while the private key decrypts it. The process ensures that only the person with the private key can decrypt the message, even though everyone knows the public key.

In 1976, Whitfield Diffie and Martin Hellman developed the Diffie–Hellman key exchange protocol—the first public key cryptography with asymmetric keys. The main strength of asymmetric encryption comes from using one-way mathematical functions because they are straightforward to calculate in the forward direction but prohibitively complex to calculate in the inverse direction. This method solves the issue of having a secure channel to share the secret because public keys are used to establish a secure channel and distribute the symmetric keys, which are then used to encrypt the messages. A digital certificate is a good example of asymmetric encryption.

Today we rely heavily on asymmetric key-based algorithms like RSA (Rivest-Shamir-Adleman), Elliptic Curve Cryptography (ECC), and DSA (Digital Signature Algorithm). These algorithms are considered safe because they rely on the difficulty of factoring large numbers or solving discrete logarithm problems. However, quantum computers can solve these problems exponentially faster than classical computers, making these encryption methods vulnerable.

Let’s examine this concept further by using an example of an RSA cryptosystem. RSA begins with the selection of two large prime numbers, p and q. These primes are multiplied to form $N = p \times q$, which becomes part of the public key. The difficulty of factoring N back into p and q is a crucial aspect of RSA's security. Another important point is to be able to calculate the periodicity. It refers to a mathematical property in modular arithmetic related to repeated patterns or cycles that can help solve the prime factorization problem. Let’s consider an example of prime number 3 (as shown in Figure 10-9) and its exponent modulo 10. You will see the values repeat after 4. So, in this case, the periodicity value is 4. RSA itself does not use period finding directly, but its security relies on the difficulty of prime factorization. You will see later how quantum computers use periodicity to break the RSA and other similar cryptography algorithms.

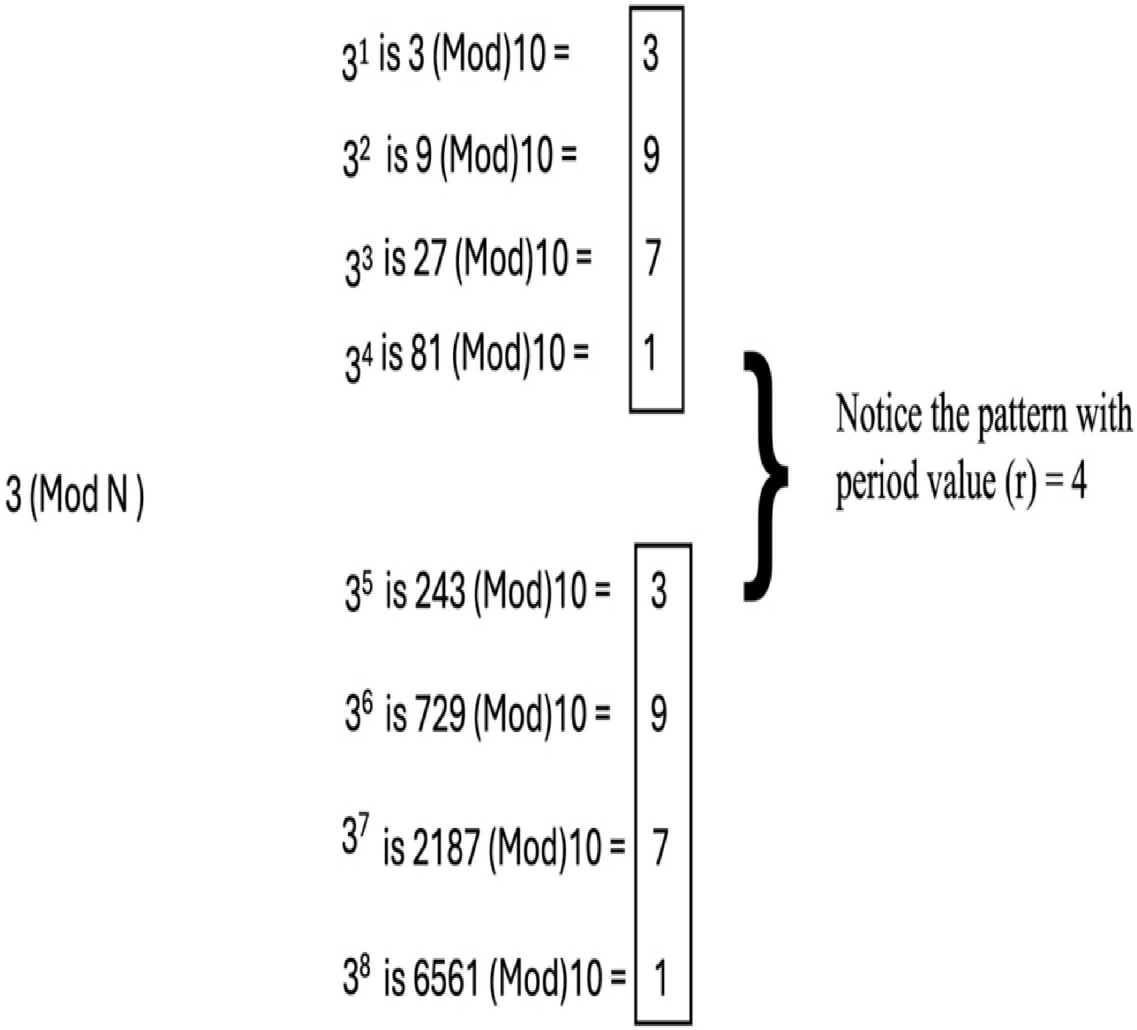


Figure 10-9 *Periodicity for Prime Number 3*

In summary, if we have two large random prime numbers, classical computers can efficiently calculate their product. However, if we are given the product of two large prime numbers, classical computers cannot efficiently factorize it to find the prime numbers p and q. Finding the periodicity r (used in algorithms like Shor's algorithm) is computationally infeasible for classical computers, but it is efficiently achievable with quantum computers, as shown in Figure 10-10.

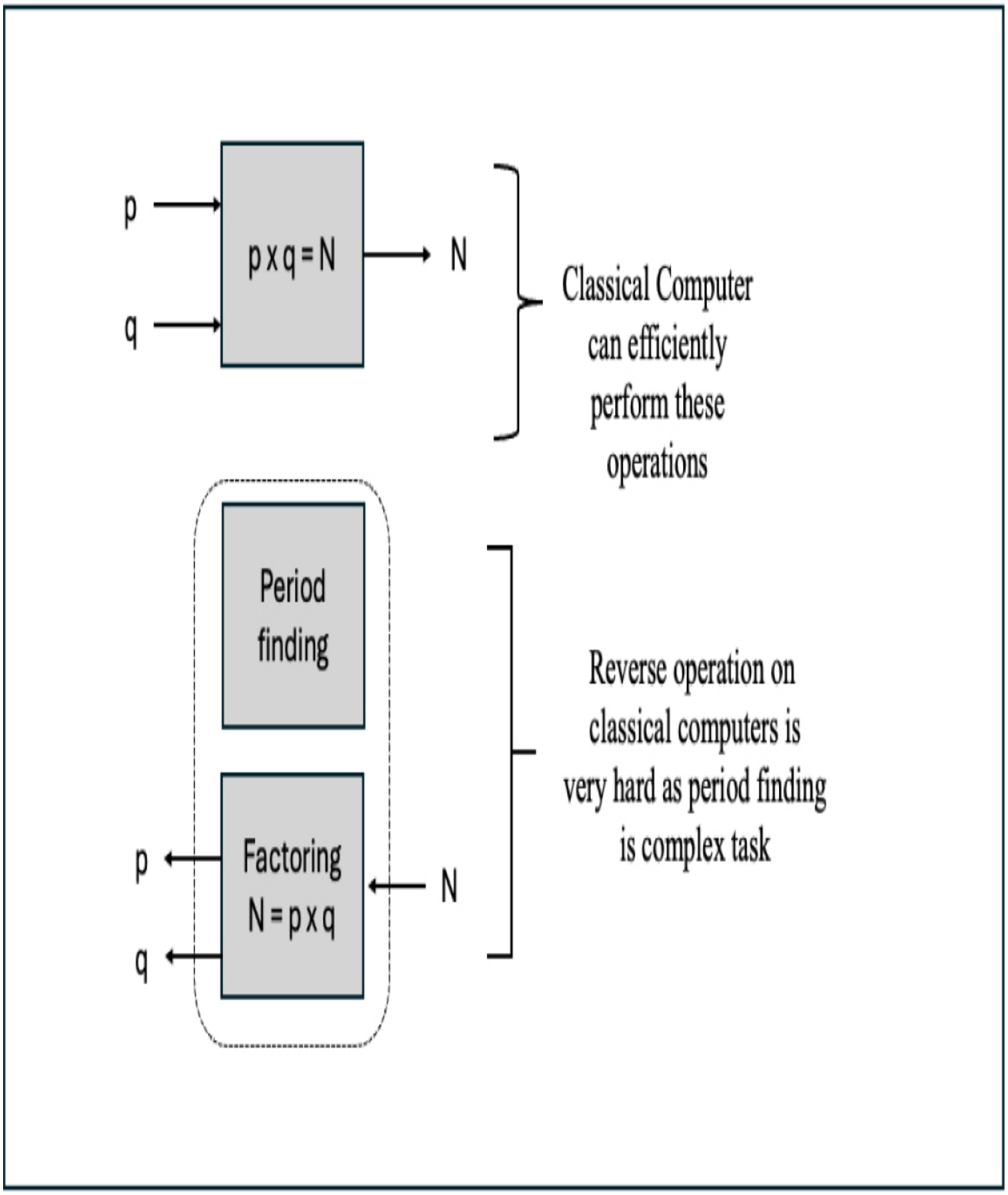

p
q
p x q = N
N
Classical Computer can efficiently perform these operations
Period finding
Factoring N = p x q
p
q
N
Reverse operation on classical computers is very hard as period finding is complex task

Figure 10-10 *Periodicity Problem with Classical Computers*

Shor’s Quantum Algorithm

Peter Shor is an American mathematician and computer scientist known for developing Shor’s algorithm. This groundbreaking quantum algorithm demonstrates how quantum computers could efficiently solve problems that are intractable for classical computers. Shor devised an algorithm that can find the prime numbers from their product. He divided the solution into two parts:

- A reduction of the factoring problem to the problem of order-finding can be done on a classical computer.
- The second bit is a quantum algorithm to solve the order-finding problem by quantum Fourier transform (QFT).

Shor's algorithm works as follows:

1. **Input:** A large composite number N, which is the product of two prime numbers, p and q. The goal is to factor N into p and q.
2. **Period Finding:** At its core, the algorithm reduces the factoring problem to a problem of finding the period of a certain function in modular arithmetic (as discussed in the previous section). Quantum computers can perform this period-finding task efficiently using quantum parallelism and quantum Fourier transform. The QFT is a critical operation used in quantum computing protocols, including period finding and phase estimation.
3. **Factoring:** Once the period is found, the algorithm can use it to calculate one of the prime factors of N, effectively breaking the encryption based on that composite number.

On classical computers, factoring a large number like those used in RSA takes exponentially more time as the number increases in size. With Shor's algorithm, a quantum computer can solve this problem in polynomial time, making it exponentially faster than classical factoring algorithms like the best-known general number field sieve.

This means that we can break the existing encryptions once we have a quantum computer with enough functioning qubits. Different companies have made great progress in this area, and there is speculation about having such a computer ready in the next decade. Such a quantum computer that is able to attack real-world cryptographic systems is termed a *cryptanalytically relevant quantum computer (CRQC)* by the National Institute of Standards and Technology (NIST).

Grover’s Algorithm

Shor’s algorithm targets asymmetric or public key cryptography systems. This means that symmetric key–based security is resistant to quantum attacks. However, as discussed earlier in the chapter, symmetric keys have their own challenges. They are also still susceptible to brute-force attacks.

Lov Kumar Grover, an Indian-American computer scientist, invented Grover’s algorithm in 1996. Grover’s quantum algorithm can search through an unsorted database much faster than classical computers. This algorithm is also capable of brute-forcing 128-bit-symmetric cryptography. Symmetric encryption algorithms, like AES, rely on the assumption that brute-forcing a key would take an impractically long time. However, Grover's algorithm reduces the complexity of brute-forcing a key from $O(2^{n)}$ to $O(2^{n/2})$, where n is the key length. To counter the effects of Grover's algorithm, cryptographers would need to double the key length. For instance, AES-256 would need to be used to maintain a security level similar to AES-128 in the quantum world. This means that symmetric encryption algorithms are a safer option than asymmetric encryption algorithms, provided we solve some of the other challenges related to management and increase the key size.

Why Worry Now?

You must be thinking, If quantum computers can break the existing cryptography and are unavailable today, why do we need to worry about securing the information? The reason is that any attacker can store encrypted information today and decrypt it once quantum computers are ready. Many organizations need to keep the information secure for 20–30 years in the future. Many government and defense deals have a contractual obligation to keep the information secure and classified for many years. What if the attackers store the information shared on the secure communication channels in the current era and use quantum technology to reach into such information a few years from now? It will reveal all the secret information. That is why you will notice that government, defense, and other organizations are actively looking at methods to deal with quantum adversaries.

Approaches to Safeguard Against Quantum Adversaries

To deal with quantum adversaries, the industry is looking at the issue from two angles:

1. **Post-Quantum Cryptography:** Researchers are working on post-quantum cryptographic algorithms that are resistant to quantum attacks to address this threat. These solutions rely on mathematical problems, like lattice-based cryptography, that quantum computers are not expected to solve efficiently. This approach includes creating new cryptography methods that will not use the prime factorization approach or other approaches that are vulnerable to quantum computers. While this chapter was being written, NIST was actively working on shortlisting some of the cryptography ciphers and has selected a couple of cryptographic schemes for further evaluation. Some of the early selection of ciphers include ML-KEM (previously know as CRYSTALS Kyber) and ML-DSA (previously known as CRYSTALS Diluthium). These are covered as part of Commercial National Security Suite 2.0 (CSNA 2.0). You can find more details about the proposed ciphers from the “CSNA Suite 2.0 and Quantum Computing FAQ” document.

2. **Quantum-Safe Strategies:** The transition to quantum-resistant encryption will be gradual but essential. Organizations are beginning to adopt quantum-safe cryptographic protocols to protect future data as quantum computers mature. This approach includes methods that can be adopted today with current cryptography to safeguard information against future quantum attackers. This includes the use of symmetric keys and their distribution through quantum methods, as discussed in the next section. In the realm of quantum network security, multiple solutions exist to address evolving threats and leverage quantum advancements. For example, symmetric keys remain widely used for encryption, but their distribution can be enhanced through quantum key distribution (QKD), which leverages the principles of quantum mechanics to securely exchange cryptographic keys between network devices. Unlike classical

methods, QKD ensures that any attempt to intercept the key is detectable, providing unprecedented security. Additionally, hybrid approaches combining QKD with existing cryptographic protocols are emerging, enabling seamless integration with current network infrastructures while preparing for post-quantum threats. These solutions provide a robust framework for securing networks in the quantum era. Multiple solutions exist in this space; you need to understand each to identify the right solution based on your use case. It is important to note that quantum-safe solutions are only for the transition phase until the new quantum-resistant ciphers are available and widely adopted in the industry. Next, let’s look at some of these solutions.

Symmetric Keys

While Grover’s quantum algorithm promises to speed up brute-force attacks, symmetric keys with sufficient length are still the first line of defense against quantum attacks. As you will notice in the next few sections, Cisco supports traditional pre-shared keys directly or via key mixing using specific algorithms on many of its product lines to allow customers to enable point-to-point quantum-safe encryption.

Quantum Key Distribution

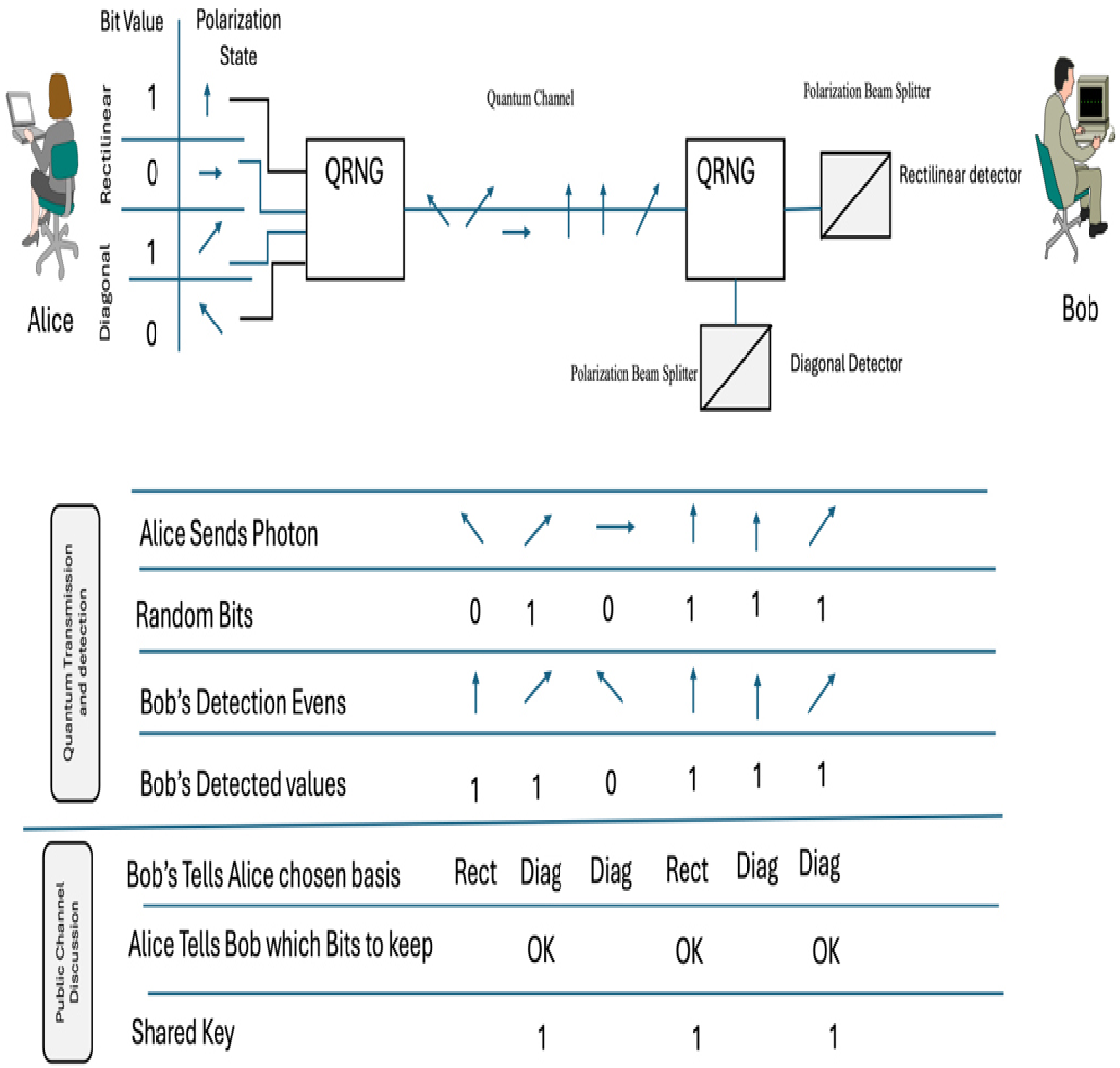
One of the challenges with symmetric keys is to share the keys between two machines in a way that cannot be compromised. Doing so is not easy using the classical communication channel, which includes IP-based communication. This is where a quantum mechanics–based key distribution method is gaining popularity. Quantum key distribution, or QKD, allows the secure sharing of key information between two devices using the quantum channel. We call it secure, because if someone tries to eavesdrop or hack into the system, it will be immediately detected, and the communication channel will stop. This is possible due to the quantum mechanics “No Cloning” theorem. In the context of QKD, it is a principle that says you cannot perfectly copy or clone an unknown quantum state, such as the quantum bits (qubits) used in QKD. This capability is crucial to the security of QKD because it prevents eavesdroppers from making a copy of the quantum data without being detected. This means that if a hacker tries to intercept and clone the qubits being used to transmit the key, they can't do it without disturbing the system. If they try to make a copy, the process will alter the qubits, alerting the legitimate parties (sender and receiver) that someone is trying to eavesdrop. There are many ways to encode information in qubits, and the mechanism for sending it across the quantum channel is known as QKD algorithms. Following are some of the common QKD protocols:

- **BB84 Protocol:** This is the first and most widely known QKD protocol. It uses photon polarization to encode bits. The sender (Alice) sends qubits to the receiver (Bob) using four possible polarization states (horizontal, vertical, and two diagonal angles). The key part is that if anyone tries to eavesdrop, they will introduce detectable disturbances because of the nature of quantum measurements. BB84 comes under the category of discrete variable protocols because the individual photon is manipulated in this approach.

- **E91 Protocol:** This protocol is based on quantum entanglement. Entangled particles are sent to two distant locations (Alice and Bob), and their measurements are correlated. If any third party tries to intercept or measure one of the particles, it disturbs the entanglement, and the eavesdropping will be detected.

- **Continuous Variable QKD (CV-QKD):** Unlike discrete-variable protocols (such as BB84), which use individual photons, CV-QKD encodes information using continuous variables like the amplitude and phase of light waves. The advantage is that CV-QKD can use standard telecom technologies like homodyne detection, making it more compatible with existing infrastructure.

Figure 10-11 shows the QKD system using the BB84 protocol.



Quantum Transmission and detection	Alice Sends Photon	↖	↗	→	↑	↑	↗
	Random Bits	0	1	0	1	1	1
	Bob's Detection Evens	↑	↗	↖	↑	↑	↗
	Bob's Detected values	1	1	0	1	1	1
Public Channel Discussion	Bob's Tells Alice chosen basis	Rect	Diag	Diag	Rect	Diag	Diag
	Alice Tells Bob which Bits to keep		OK		OK		OK
	Shared Key		1		1		1

Figure 10-11 *QKD System with BB84*

One of the main challenges with QKD is that it requires a separate quantum channel for its operations. This means a dedicated dark fiber between two points, such as the head office and a branch site. At the time of writing this chapter, this was limited to a few hundred kilometers. This constraint limits the use of QKD to nearby devices (typically up to 150–200 km or up to 130 miles). You cannot use the repeaters because it will go against the No Cloning theorem, as discussed previously.

Practical Solution Approach

One of the common questions from customers is where they can start if they need to adopt security measures against quantum adversaries. Our answer is that every customer should evaluate the risks of quantum technology based on their requirements. They should include quantum threats in their security policies and start evaluating solutions from the vendors. This is an evolving area of technology, and new solutions keep coming. You can start by answering some of these questions:

- Which of my organization's data and communications are most at risk from quantum attacks?
- What are the potential risks if I delay implementing quantum-safe solutions, the potential for adversaries to steal encrypted data today, and decrypt it later with quantum computers?
- How is my network currently secured, and which cryptographic protocols are we using (e.g., RSA, ECC, AES)?
- What is the best way to introduce quantum-safe technologies with minimal downtime or disruption to my business?
- Are my vendors and partners taking steps toward quantum-safe security?
- How much will it cost to implement quantum-safe solutions, and what resources will I need (e.g., specialized personnel, new tools)?
- Are there industry standards, regulations, or compliance frameworks that I need to follow when adopting quantum-safe solutions?
- How will I keep up with advancements in quantum computing and quantum-safe security over time?

NIST is still in the process of selecting and finalizing the post-quantum cryptography candidates. It will be a while before the various networking vendors finalize and implement such a solution. You should consider some of the solutions available today to safeguard critical links. Cisco closely follows the NIST guidelines to include post-quantum ciphers in its products. Cisco also supports a variety of solutions for the transition phase. Let’s look at some of the solution approaches from Cisco; keep in mind that the end goal for your organization must be to adopt the quantum-safe cryptography natively on the networking products and solutions you deploy.

You may add an additional layer of security by using QKD or similar quantum mechanism-based solutions like quantum entropy generators.

Cisco supports quantum-resistant IPsec on its IOS-XE and quantum-resistant MACsec on its IOS-XR platforms.

Quantum-Safe IPsec

Cisco IOS-XE supports the quantum-resistant IPsec tunnel from software version 17.11. There are primarily two methods for creating it:

1. **Manual Post-Quantum Pre-Shared Keys (PPK):** This is the simplest method to deploy quantum-safe IPsec tunnels without the requirement of any additional third-party hardware or the need to understand the quantum devices. This approach is based on the symmetric key's resistance to quantum adversaries. However, the pre-shared keys are not used directly. Instead, they use a key mixing mechanism as defined in RFC 8784. It is an extension of the IKEv2 protocol. This feature applies to all IKEv2 and IPsec VPNs, such as FlexVPN (SVTI-DVTI) and DMVPN, except for GETVPN. This manual method is not very scalable. Still, it serves the purpose where there are limited connections to be secured, and organizations are trying the quantum solution in a restricted environment. The PPK used must be of sufficient size and entropy. It is also recommended that you rotate the keys often. Figure 10-12 shows the manual PPK method, and Figure 10-13 shows the output from the router CLI after establishing quantum-resistant IPsec.

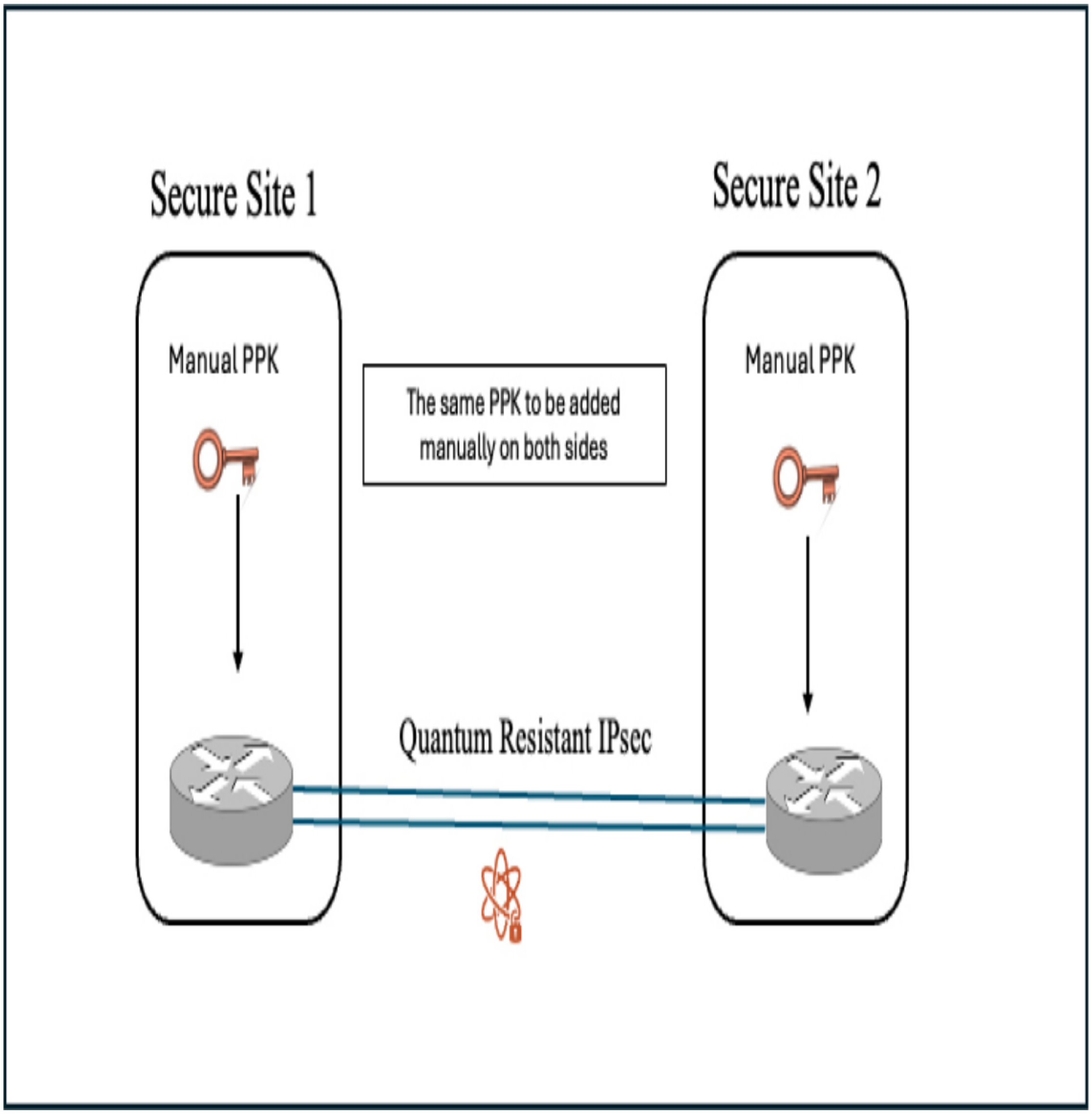
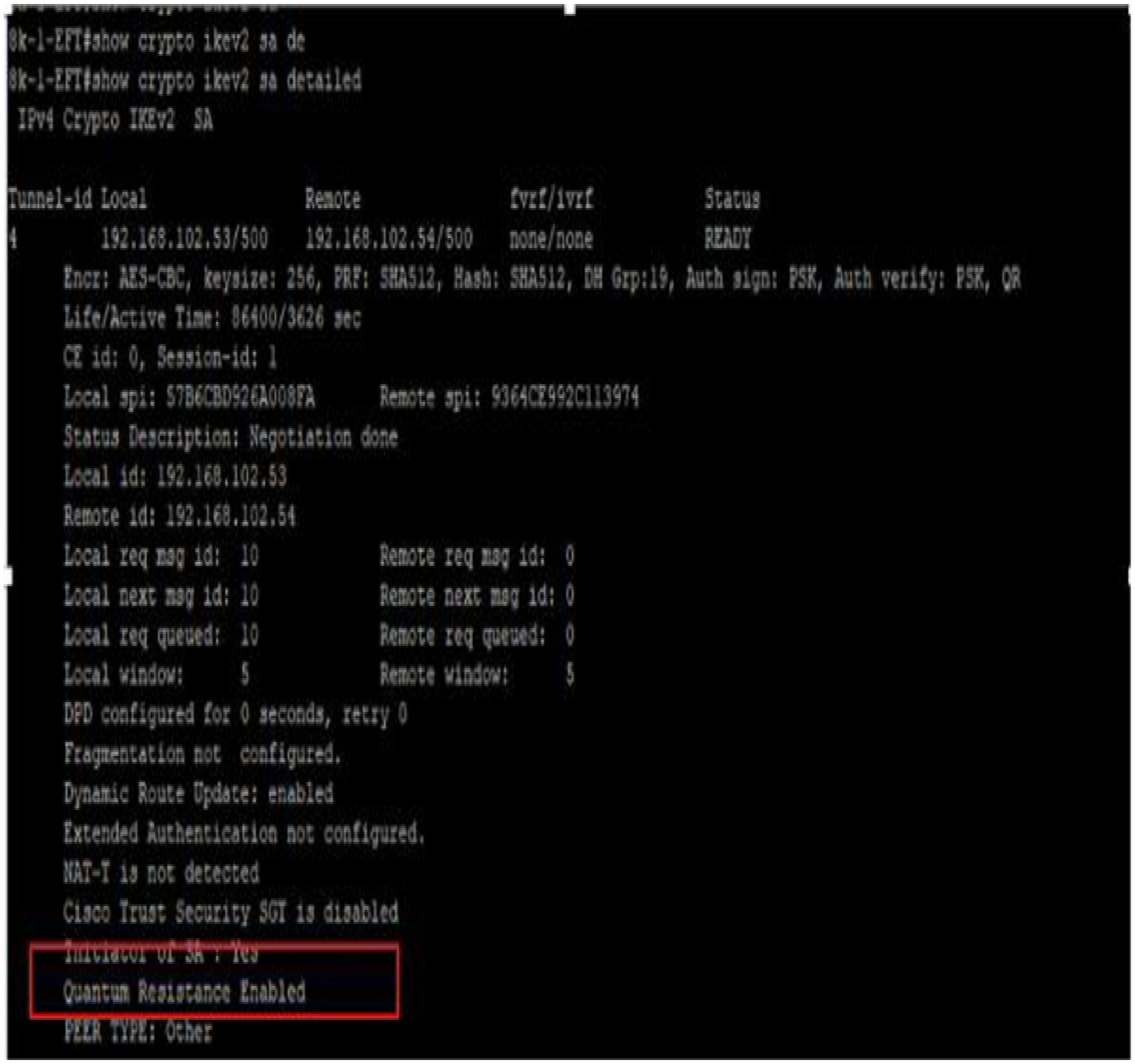


Figure 10-12 *Manual PPK-Based Quantum-Resistant IPsec*



```
8k-1-EFT#show crypto ikev2 sa de
8k-1-EFT#show crypto ikev2 sa detailed
 IPv4 Crypto IKEv2  SA

Tunnel-id Local                Remote               fvrf/ivrf            Status
4         192.168.102.53/500   192.168.102.54/500   none/none            READY
      Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: PSK, Auth verify: PSK, QR
      Life/Active Time: 86400/3626 sec
      CE id: 0, Session-id: 1
      Local spi: 57B6CBD926A008FA       Remote spi: 9364CE992C113974
      Status Description: Negotiation done
      Local id: 192.168.102.53
      Remote id: 192.168.102.54
      Local req msg id:  10             Remote req msg id:  0
      Local next msg id: 10             Remote next msg id: 0
      Local req queued:  10             Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      Fragmentation not  configured.
      Dynamic Route Update: enabled
      Extended Authentication not configured.
      NAT-T is not detected
      Cisco Trust Security SGT is disabled
      Initiator of SA : Yes
      Quantum Resistance Enabled
      PEER TYPE: Other
```

Figure 10-13 *Sample Output Showing Quantum-Resistant IPsec on IoS XE*

2. **Dynamic Post Quantum Pre-Shared Keys with QKD Devices:** As we know, Manual PPK configuration has several issues related to key management, such as rotation and entropy. To solve this issue, Cisco IOS-XE also supports integration with third-party entropy or key providers. These systems could be quantum, allowing QKD-based key generators to be integrated with Cisco routers. In such a case, integration requires Cisco Secure Key Integration Protocol (SKIP) support on both key providers and Cisco routers. SKIP is an HTTPs-based protocol, and any QKD/key entropy vendor can implement it. Figure 10-14 explains the SKIP integration with the PPK source.

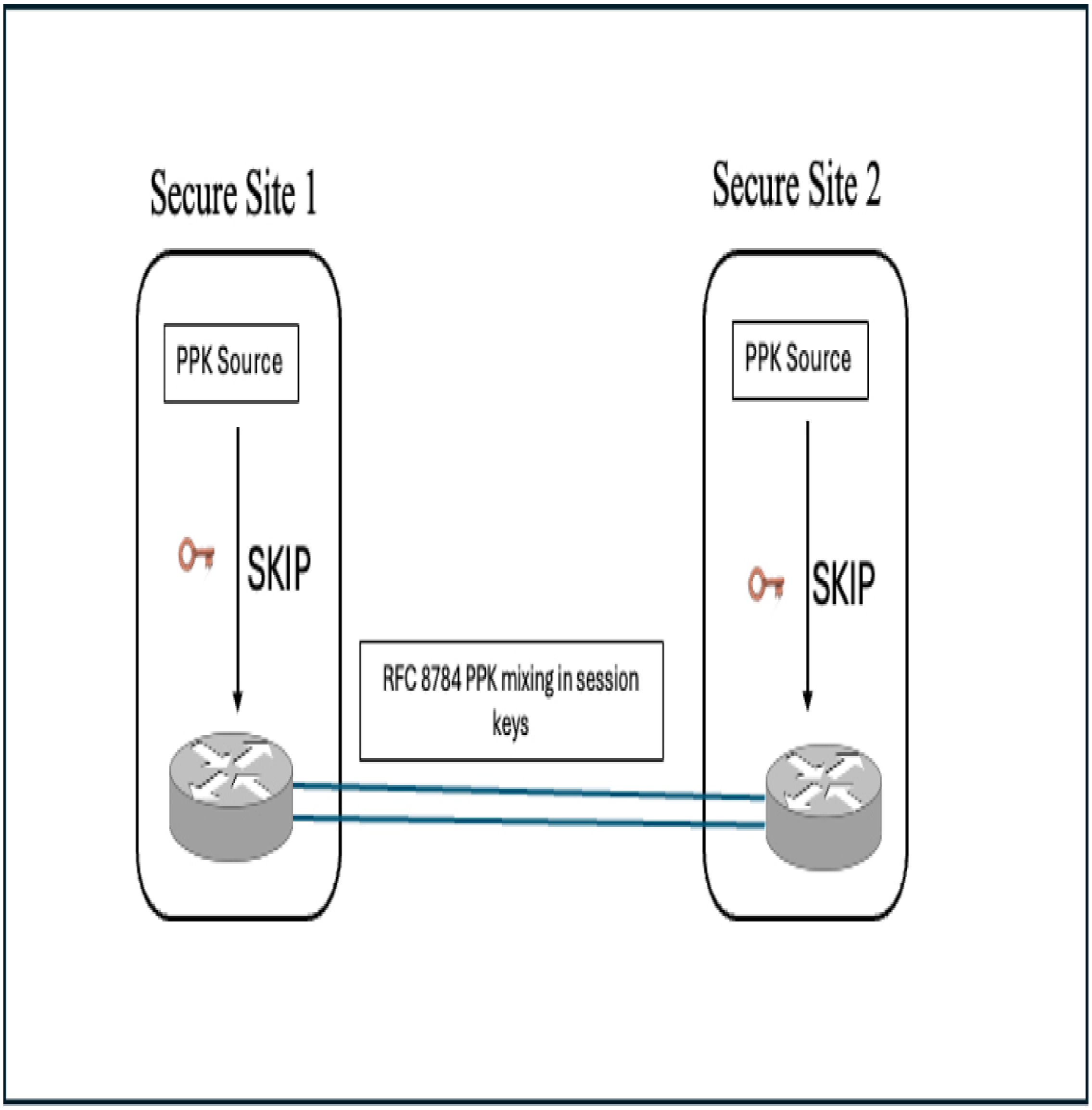


Figure 10-14 *SKIP-Based Integration with Key Source*

An ETSI-based integration mechanism also integrates with the external QKD devices. At the time of writing this chapter, this method is not supported on Cisco devices.

A common implementation of dynamic PPK is integration with QKD devices, as shown in Figure 10-15. These devices are quantum based and use their dedicated quantum channel to generate key entropy. From a Cisco device perspective, PPK is pulled using the SKIP protocol from the QKD device. This method is definitely better than manual PPK because it uses quantum properties to generate symmetric keys at both ends. Assuming the QKD vendor has implemented a solution based on known quantum principles, the No Cleaning theorem will ensure that the eavesdropper cannot capture the keys or keying material. This allows you to have symmetric keys at both ends of the IPSec tunnel without sending the real keys. This is the beauty of quantum technology. However, it still has a few limitations:

- Current QKD devices cannot function over 100 km due to limitations in the ability to keep the quantum state usable over longer distances over existing fiber technology.
- This solution is expensive because you will need a dedicated QKD box at each end of the tunnel.

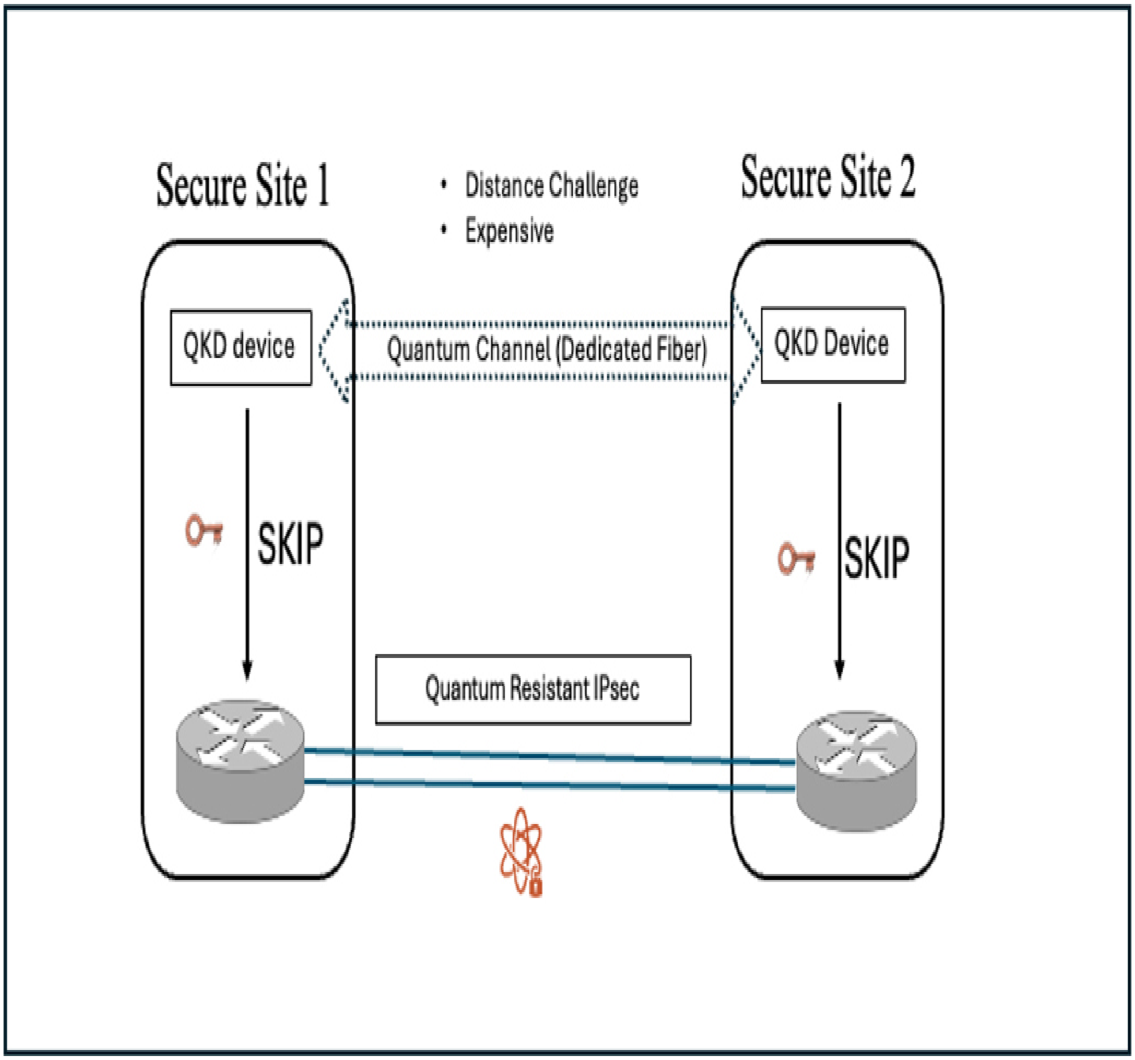


Figure 10-15 *SKIP-Based Integration with QKD Device*

A few QKD vendors offer quantum entropy as a service via cloud platforms. In such cases, you must ensure the security of key delivery mechanisms.

Quantum-Safe MACsec

Cisco IOS-XR devices support quantum-resistant MACsec on different platforms. Please refer to the Cisco documentation for exact details on hardware, software, and virtual platforms. Quantum-resistant MACsec works similarly to IPsec and supports manual and dynamic key solutions.

Manual Pre-Shared Key: Existing methods of using manual pre-shared keys are resistant to quantum attacks, provided they are of sufficient length and entropy. It is important to note that this method does not use key mixing, as suggested by RFC 8784. This is a traditional PSK solution. Like any other pre-shared key solution, this method has the same limitations related to key management. Figure 10-16 shows quantum-resistant MACsec with PSK.

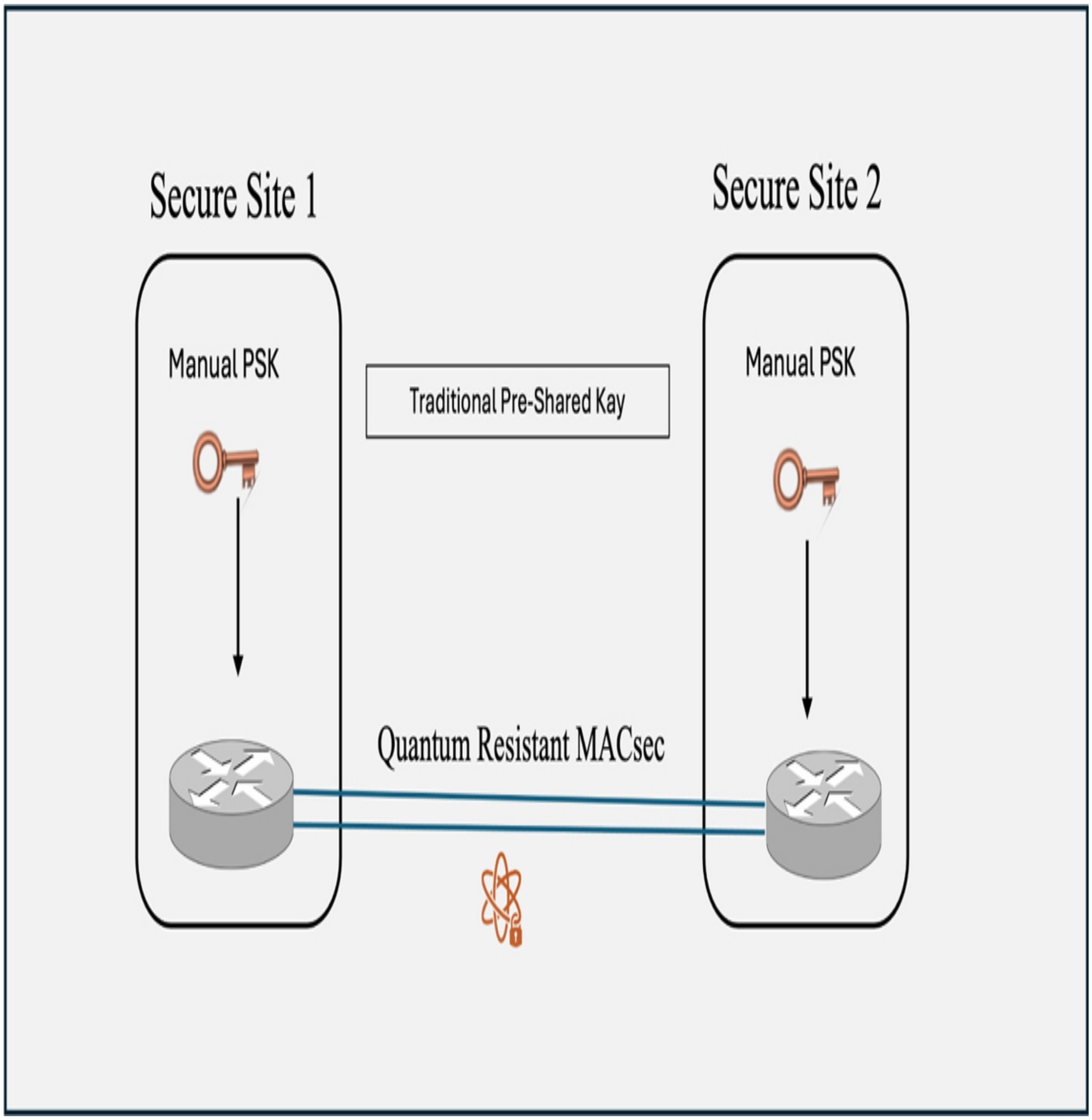


Figure 10-16 *Quantum-Resistant MACsec with PSK*

Dynamic Keys

MACsec on Cisco IOS-XR supports two different methods of dynamic keying:

1. **QKD Integration via SKIP:** This method is similar to the IPsec solution, where integration with external QKD key providers is supported using the SKIP protocol. In this case, the limitation of distance and additional devices still remains. Figure 10-17 shows the QKD integration for MACsec on Cisco IOS-XR.

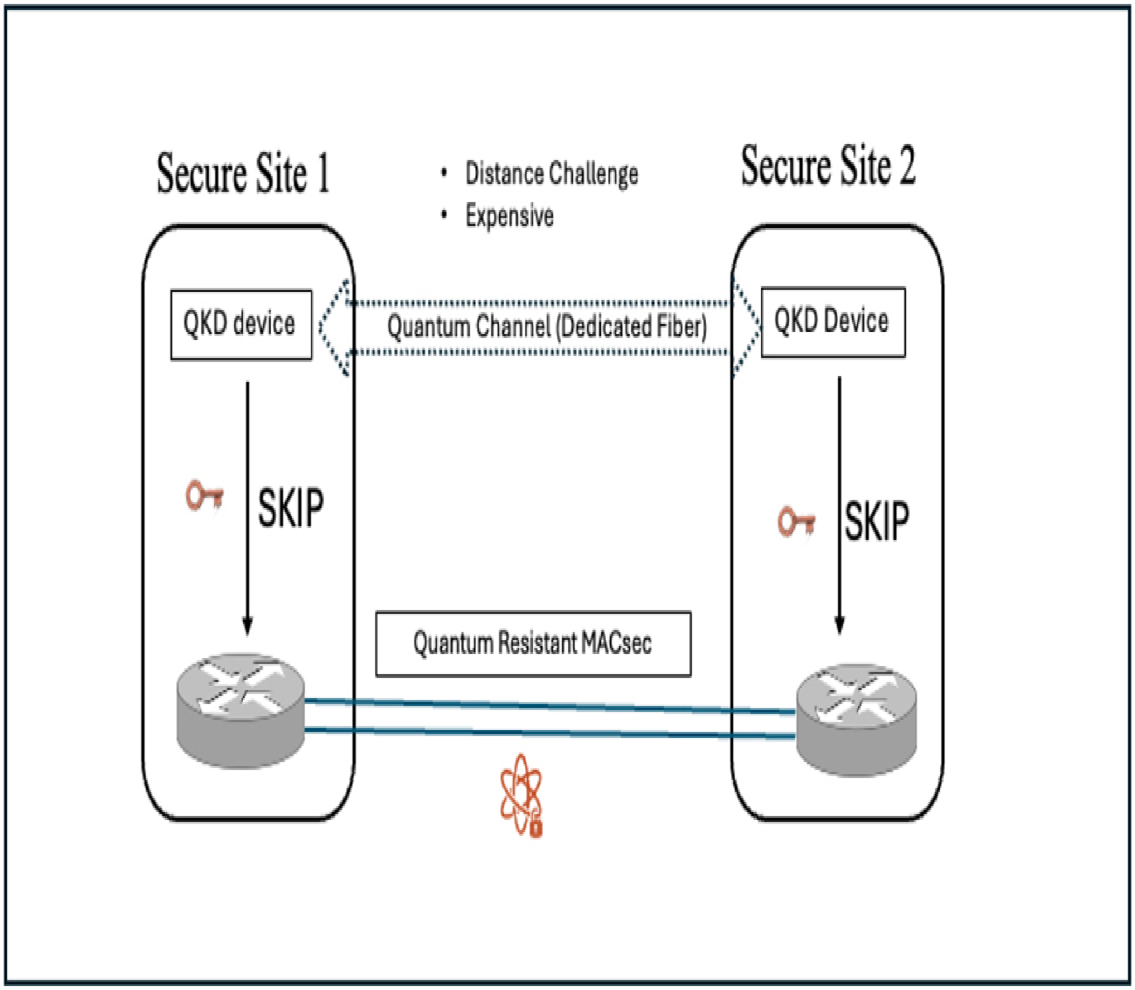


Figure 10-17 *Quantum-Resistant MACsec with QKD Integration Using SKIP*

2. **Software Entropy:** IOS-XR also supports software-based entropy generation and key distribution, as shown in Figure 10-18. In this case, a dedicated key generation module (Cisco SKS) is available as a command line, and you can use the same for dynamic symmetric key generation. This method is not as secure as external QKD. Still, it allows customers to try quantum-safe masses without limitations like distance and cost of previous QKD device-based methods, and it is completely free. Key distribution is based on the McEliece cryptosystem, which is considered quantum safe. The core challenge in breaking McEliece lies in the computational difficulty of decoding random linear codes, even when a partial code is given. This problem remains complex even for quantum computers because it doesn’t reduce easily to problems like factoring or discrete logarithms, which are vulnerable to quantum algorithms. As such,

McEliece is one of the promising candidates for post-quantum cryptography, offering long-term security against both classical and quantum computers. However, the main drawback is that it requires large key sizes, which can make it less practical in some environments compared to other cryptosystems.

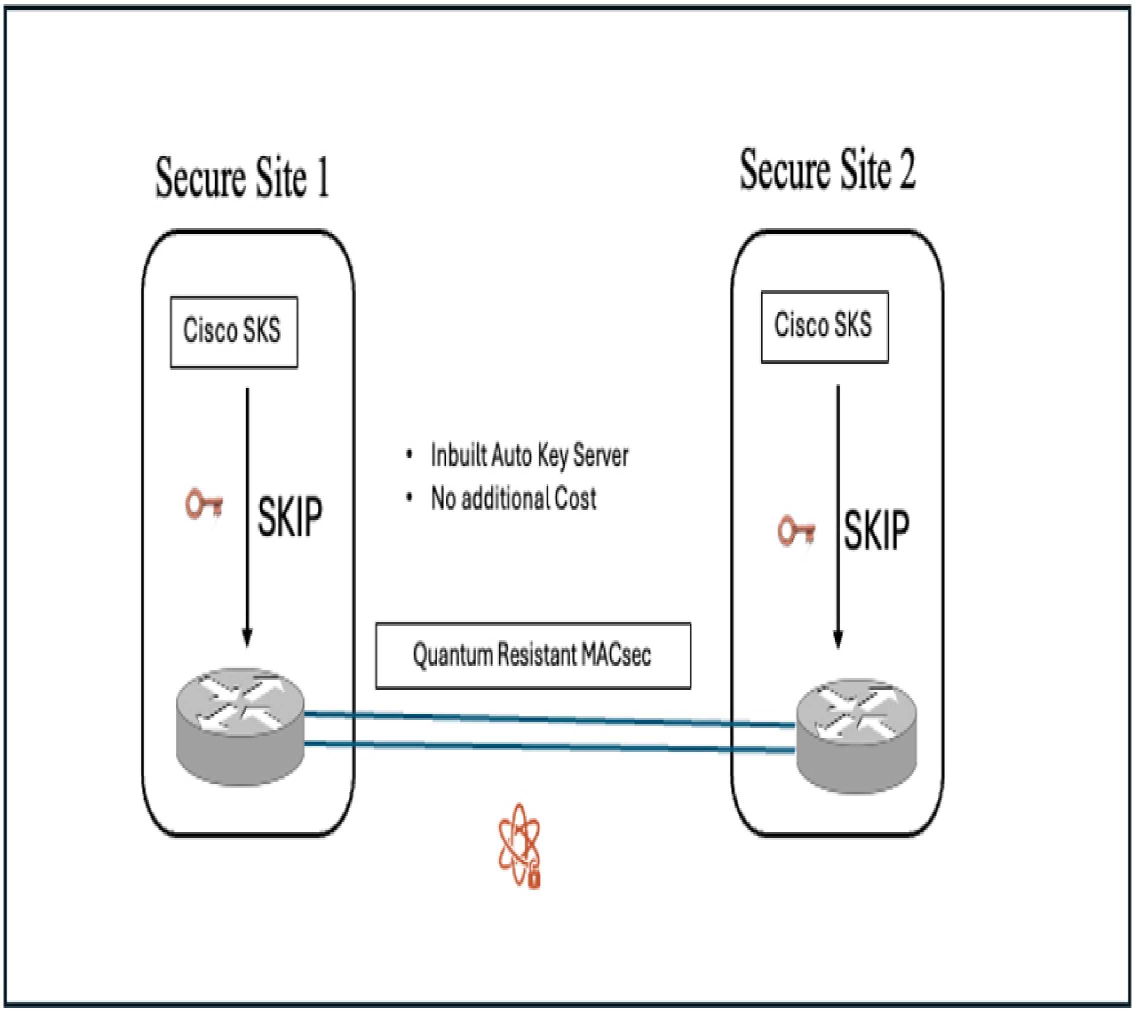


Figure 10-18 *SKIP-Based Integration Between IOS-XR and Software Key Source*

Again, the aforementioned solutions are transitionary to native quantum-safe cryptography. Different vendors offer new solutions and approaches every few months. As a security architect, you need to be aware of all developments in quantum security and continuously update your security strategy.

Summary

In this chapter, you learned about the fundamentals of quantum computing and how a quantum computer uses qubits as base elements for calculations. It is important to remember that quantum computers are not faster and more innovative versions of classical computers; instead, they represent a new way of computing based on the principles of quantum mechanics. You also learned how quantum computers can pose risks to existing cryptography and the various methods to safeguard information from quantum adversaries. Please note that quantum computing is an evolving field, with new progress being made every month. Keep yourself updated about NIST quantum-safe cryptography, vendors' implementation, and progress made by vendors offering QKD solutions, existing cryptography, and the various methods to safeguard information from quantum adversaries.

Chapter 11. Network Convergence and Considerations

In this chapter, you will learn about the following:

- Convergence in classic IP-routed networks
- Convergence in software-defined networks
- The impact of convergence on security protocols
- Methodologies in convergence testing
- How to simulate convergence scenarios
- How to monitor convergence of TrustSec and security enforcement

What Is Convergence?

The topic of Layer 3 routing convergence is something that historically has been considered a network protocol challenge. Convergence is normally considered the ability for a routing table to repopulate with the right next hop and forwarding information derived from the link state and distance vector databases to create the right decisions. These decisions are manifested through the update of the routing table to reflect changes in the network topology or routing paths. While this is a true aspect of how network convergence was originally defined, further considerations go beyond simply populating a routing table with accurate and up-to-date information.

At a fundamental level, the routing and forwarding instances configured and in use on physical and virtual network devices represent the foundation on which convergence needs to take place. Advanced protocols that often work in parallel to the routing protocol in the underlay network, "ride on top" of the base control plane and forwarding paths that are configured within the platforms.

For example, in a routed network, IP protocols such as Border Gateway Protocol (BGP) are often reliant on communications being established between peers; in scenarios with peering between loopback addresses exist, this can often have a further dependency on the Internal Gateway Protocol (IGP) such as Open Shorted Path First Protocol (OSPF), which is configured in the underlay of the network (the global routing table).

Figure 11-1 shows a point-to-point link where BGP does not have any specific requirement on other IGPs, meaning that protocol communications that facilitate convergence for BGP can be achieved without additional dependencies on other protocols.

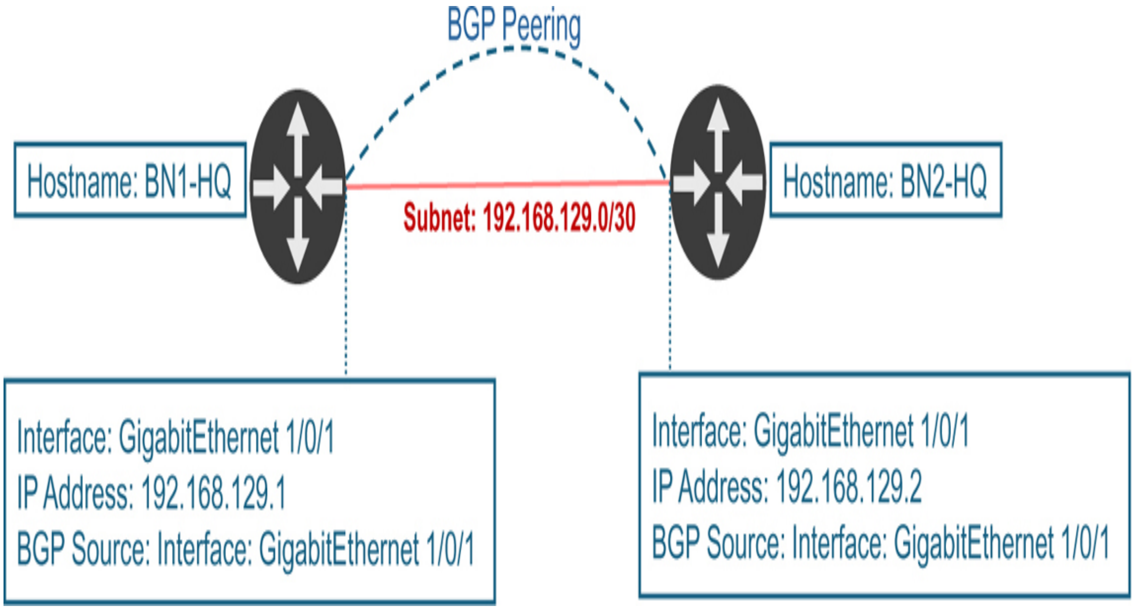
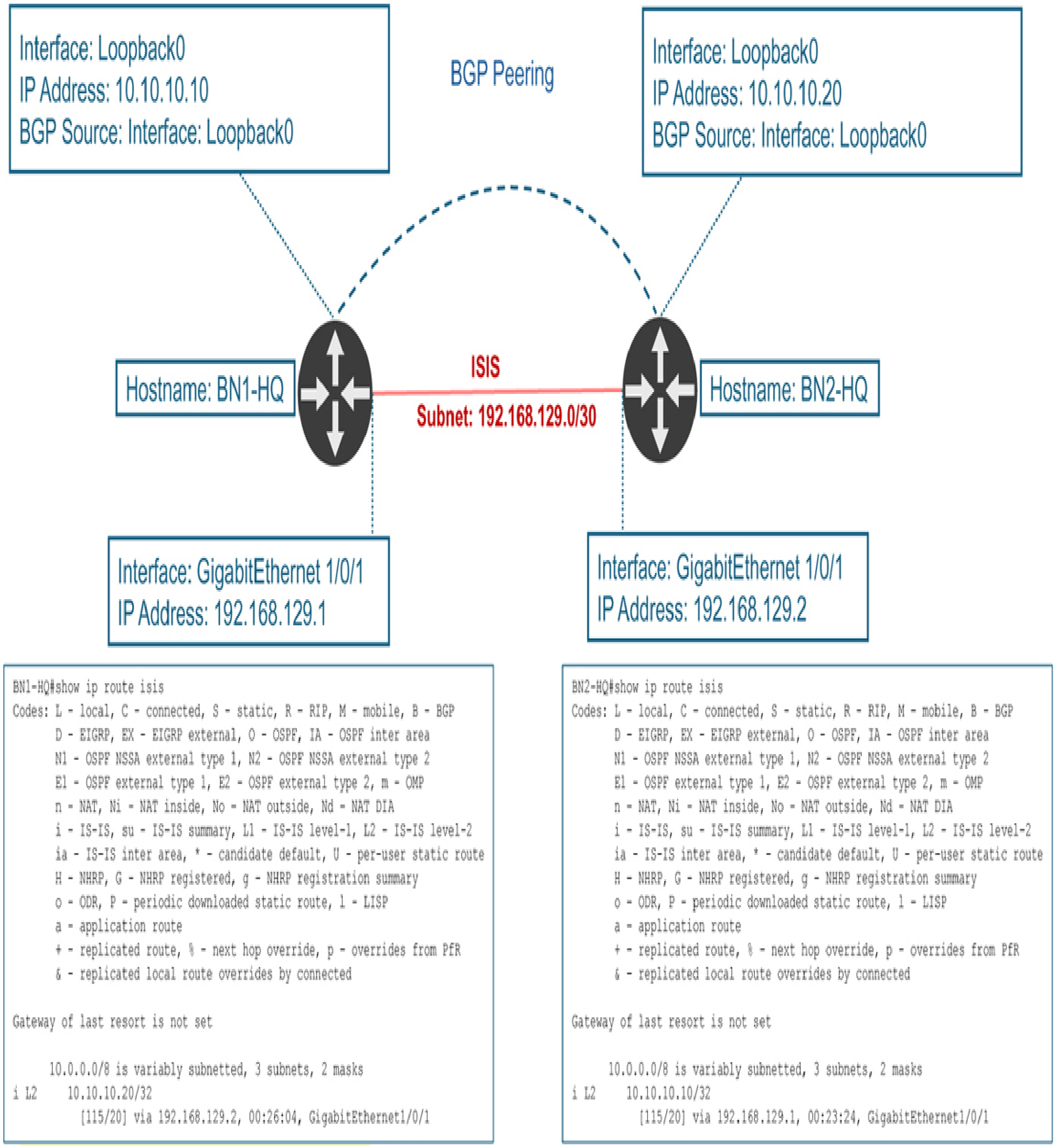

BGP Peering
Hostname: BN1-HQ
Hostname: BN2-HQ
Subnet: 192.168.129.0/30
Interface: GigabitEthernet 1/0/1
IP Address: 192.168.129.1
BGP Source: Interface: GigabitEthernet 1/0/1
Interface: GigabitEthernet 1/0/1
IP Address: 192.168.129.2
BGP Source: Interface: GigabitEthernet 1/0/1

Figure 11-1 *BGP Peering: Using Physical Interfaces*

Figure 11-2, on the other hand, has a dependency on the Intermediate System to Intermediate System (IS-IS) routing protocol to ensure that the loopback addresses that are in use (10.10.10.10/32 and 10.10.10.20/32) are reachable between the two nodes. Although achieving convergence in this scenario could be considered relatively seamless and straightforward, further dependency on additional protocols is a consideration for you to anticipate and understand when attempting to measure overall convergence.


Interface: Loopback0
IP Address: 10.10.10.10
BGP Source: Interface: Loopback0
BGP Peering
Interface: Loopback0
IP Address: 10.10.10.20
BGP Source: Interface: Loopback0
Hostname: BN1-HQ
ISIS
Subnet: 192.168.129.0/30
Hostname: BN2-HQ
Interface: GigabitEthernet 1/0/1
IP Address: 192.168.129.1
Interface: GigabitEthernet 1/0/1
IP Address: 192.168.129.2

```
BN1-HQ#show ip route isis
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
i L2     10.10.10.20/32
           [115/20] via 192.168.129.2, 00:26:04, GigabitEthernet1/0/1
```

```
BN2-HQ#show ip route isis
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, m - OMP
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       H - NHRP, G - NHRP registered, g - NHRP registration summary
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR
       & - replicated local route overrides by connected

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
i L2     10.10.10.10/32
           [115/20] via 192.168.129.1, 00:23:24, GigabitEthernet1/0/1
```

Figure 11-2 *BGP Peering Using Source Loopback*

In this chapter, we will look into convergence factors as a whole, from a network communications perspective, as well as from a security perspective, in the spirit of maintaining a zero trust network architecture and the carriage and maintenance of microsegmentation-based identifiers such as security group tags (SGTs).

Convergence in Layer 3 Routed Architectures

In Layer 3 routed networks, convergence scenarios have dependencies on a number of factors, ranging from the time to detect the failure of a link or circuit to protocol hold times and the time to process large routing updates. In addition to the convergence of the network that may be representing the underlay (the global routing table), there is also the need to converge protocols that may ride atop the underlay, such as multiprotocol BGP and VPNv4/VPNv6 label tables in MPLS networks, protocols that may be running within tunnels such as IPSEC or GRE, or further control protocols that may exist in overlay networks.

Over the years, various capabilities have been introduced into routing protocols; the purpose was to improve scenarios to either more rapidly detect a fault or loss of connectivity or more gracefully converge, limiting the time that traffic may be blackholed and dropped as a result of taking a stale or a no-longer-viable path. Capabilities included protocol synchronization features that allow dependent protocols to wait for one another, such as MPLS LDP IGP synchronization that more commonly exists in service provider domains. They also included the use of the IS-IS overload bit, which allows for the selective removal of network routers from the forwarding path (prior to maintenance activities), as shown in Figure 11-3. Also, they included conditional routing through the use of configuration levers like non-exist and advertise-maps in BGP. These capabilities often provided a way to optimize network reliability to lower the impact that could occur during the loss of a redundant network node.

Protocols such as BGP and the IGP that is selected for use serve the purpose of ensuring that client-to-server communications take the appropriate path through the network from source to destination. This path is important in the context of microsegmentation because, depending on the topology, inline tagging may be used; it requires forwarding over specific links that are capable of maintaining the adaptive policy or Cisco metadata header that is carrying the tag. Certain line cards, platforms, or network modules may not be capable of maintaining the tag and subsequently drop the information, leading to incorrect or missing tag allocations for the traffic.

The routing protocols also serve an important purpose when it comes to ensuring that serving instances of control traffic for security protocols (such as TACACS+, RADIUS, and the SXP protocol that can be used for transmitting or receiving IP-to-SGT binding information) are reachable and can therefore properly serve the network with the most up-to-date authorization, policy, or binding states.

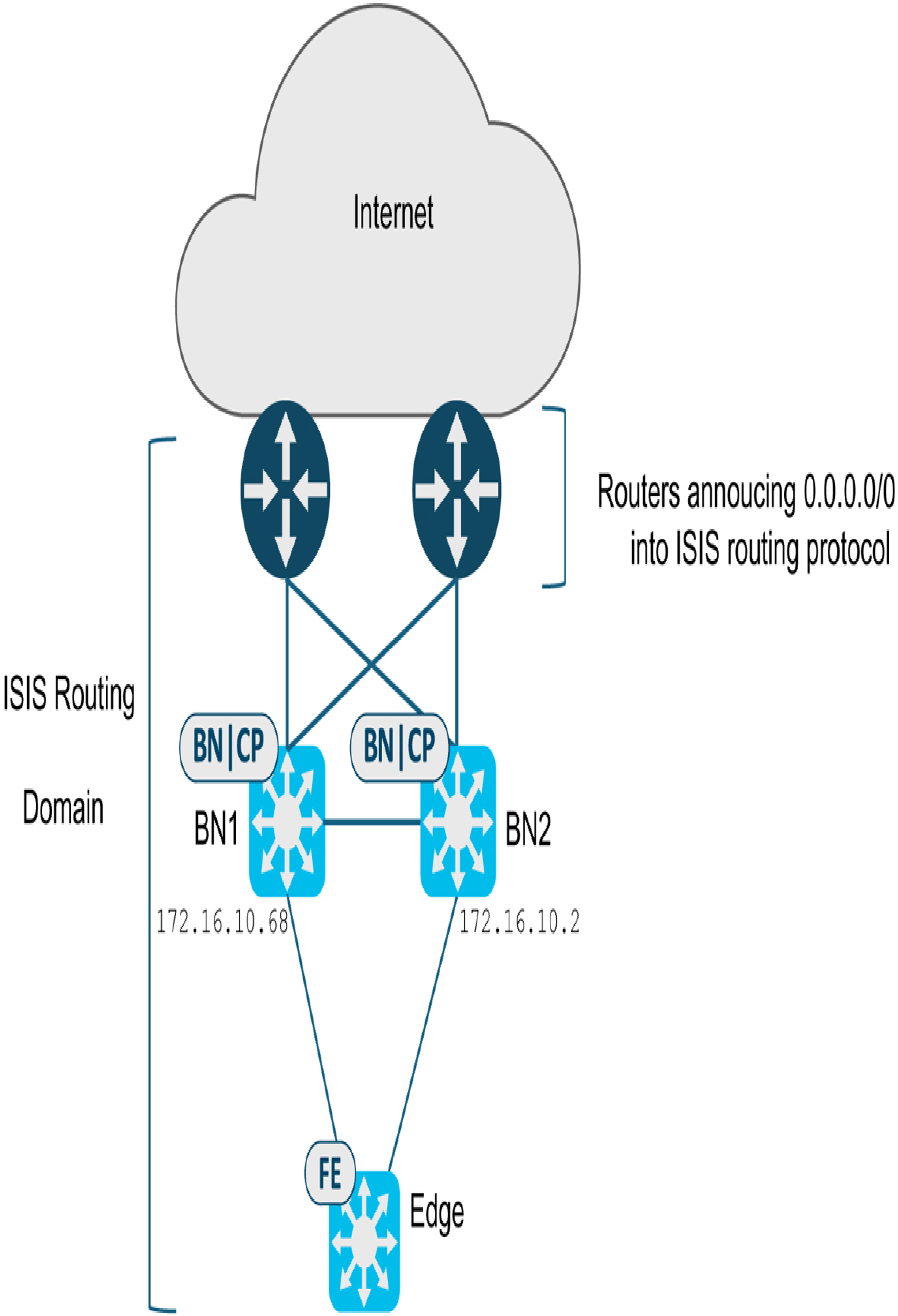

Internet
Routers annoucing 0.0.0.0/0
into ISIS routing protocol
ISIS Routing
Domain
BN|CP
BN|CP
BN1
BN2
172.16.10.68
172.16.10.2
FE
Edge

Figure 11-3 *IS-IS Routing Topology*

Example 11-1 through Example 11-3 showcase the behavior seen from the perspective of the Edge switch in Figure 11-3, where the IS-IS overload bit is set, opting out of the secondary forwarding path and allowing maintenance activities to be conducted on the redundant border without impacting traffic that may be traversing from the edge node in the global routing table.

Example 11–1 *Change of Routing on the Edge Device Upon Changing the IS-IS Overload Bit on the Upstream Border*

```
Edge#show ip route super
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external ty
       E1 - OSPF external type 1, E2 - OSPF external type 2, m -
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS
       ia - IS-IS inter area, * - candidate default, U - per-user
       H - NHRP, G - NHRP registered, g - NHRP registration summa
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides
       & - replicated local route overrides by connected

Gateway of last resort is 172.16.10.68 to network 0.0.0.0

i*L2  0.0.0.0/0 [115/10] via 172.16.10.68, 00:00:08, GigabitEther
                [115/10] via 172.16.10.2, 00:00:08, GigabitEthern
Edge#

BN2-HQ#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
BN2-HQ(config)#router isis
```

```
BN2-HQ(config-router)# set-overload-bit
```

Example 11–2 *Resulting Debug Output of Route Withdrawal on the IS-IS Overload Bit Being Set Up on BN2-HQ Node*

```
Edge#debug ip routing
IP routing debugging is on
Edge#term mon
Edge#
2290324: Mar 17 13:54:47.430: RT: del 0.0.0.0 via 172.16.10.2, is
2290325: Mar 17 13:54:47.430: 'isis' add_route, default(0x0):0.0.
tracking 0 type 0x80 subnet_masklen 0 omp_tag 0x0
2290326: Mar 17 13:54:47.430: nh-1, topoid 0x0 flags 0x0 attr 0x1
172.16.10.65 gw_v6 :: source_v6 :: out_sid_v6 :: metric 10 tag 0
0x100001 handle 0/0x0
2290327: Mar 17 13:54:47.430: RT: updating isis 0.0.0.0/0 (0x0) o
    via 172.16.10.68 Gi1/0/1  0 0 0x0 1048578 0x100001
Edge#
```

Example 11–3 *Updated Routing Table Entry Showing Reduced Forward Paths Available for the Default Route*

```
Edge#show ip route super
Codes: L - local, C - connected, S - static, R - RIP, M - mobile,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external ty
       E1 - OSPF external type 1, E2 - OSPF external type 2, m -
       n - NAT, Ni - NAT inside, No - NAT outside, Nd - NAT DIA
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS
       ia - IS-IS inter area, * - candidate default, U - per-user
       H - NHRP, G - NHRP registered, g - NHRP registration summa
       o - ODR, P - periodic downloaded static route, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides
       & - replicated local route overrides by connected

Gateway of last resort is 172.16.10.68 to network 0.0.0.0

i*L2  0.0.0.0/0 [115/10] via 172.16.10.68, 00:00:06, GigabitEther
Edge#
```

More elaborate levers sometimes need to be considered when looking at conditional advertisements of networks, based on the use of trigger prefixes, to permit announcement.

The criticality in time required to converge can be highly dependent on factors at play within the organization that is running the network or providing the service and the dependent applications. If the applications have a very low tolerance to disruption and packet loss, then more aggressive convergence timers may be required.

In domains such as merchant banking, healthcare, and high frequency trading, applications are very unforgiving, whereas other environments may have much less stringent application requirements.

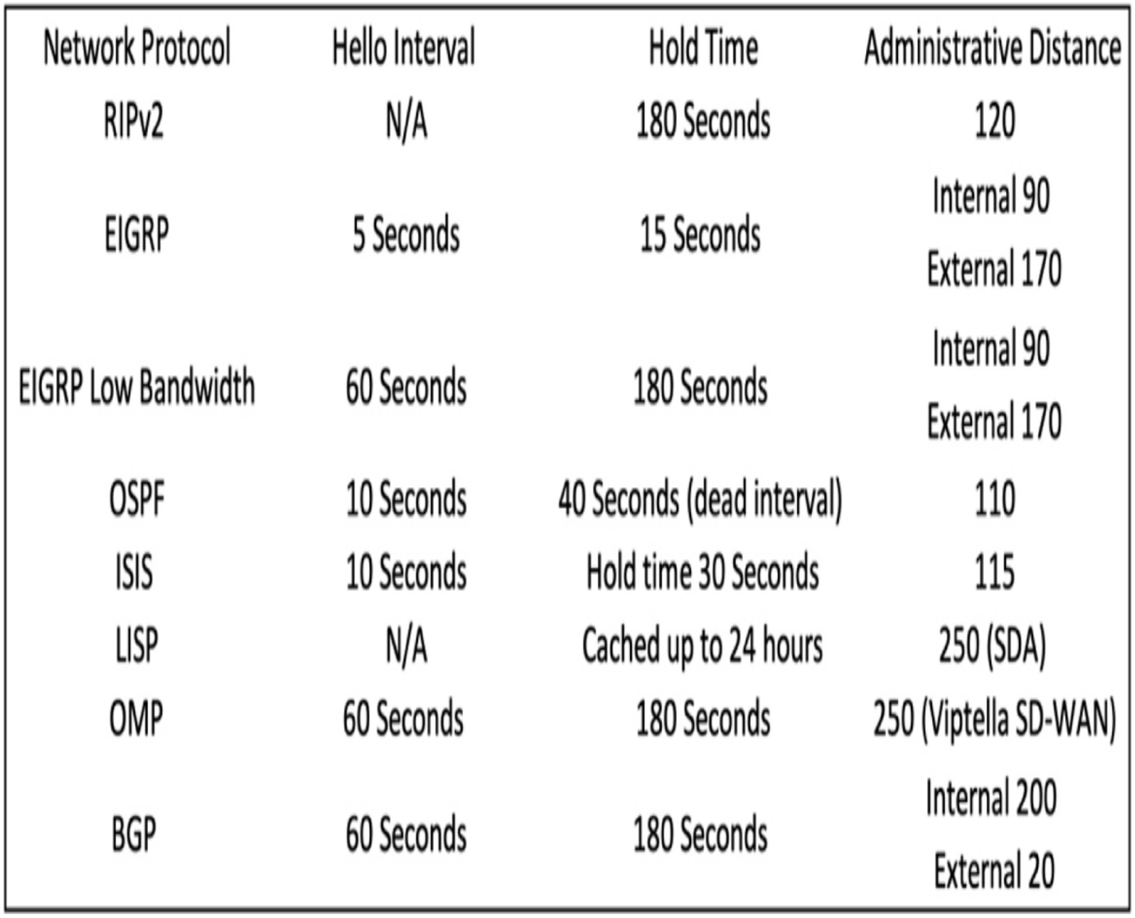
For enterprise environments, based on the experience of solution validation testing teams within Cisco, generally the following noninteractive application handling is seen during convergence-induced loss:

- Traffic loss of up to 3 seconds usually does not impact many enterprise applications.
- Traffic loss of 3–5 seconds can result in application transaction failures.
- Traffic loss of 5 seconds tends to result in hard resets of applications.

Interactive applications, such as real-time voice, however, tend to have very strict latency requirements with recommendations of up to 300 ms round-trip to avoid noticeable disruption. These sorts of traffic flows often tend to lead to enterprises that are adopting features and functionalities into their environments supporting more rapid convergence.

Protocol Convergence Timers

The convergence time of protocols, as depicted in Figure 11-,4 often depends on the timers associated with their speed to detect the loss of communications to a particular network. These timers have the consequence of maintaining a network in their routing information base (routing table) that may have been learned through multiple protocols in parallel. The deal-breaker between protocol selection in such a scenario, assuming that the network uses the same length subnet mask, is the administrative distance field.

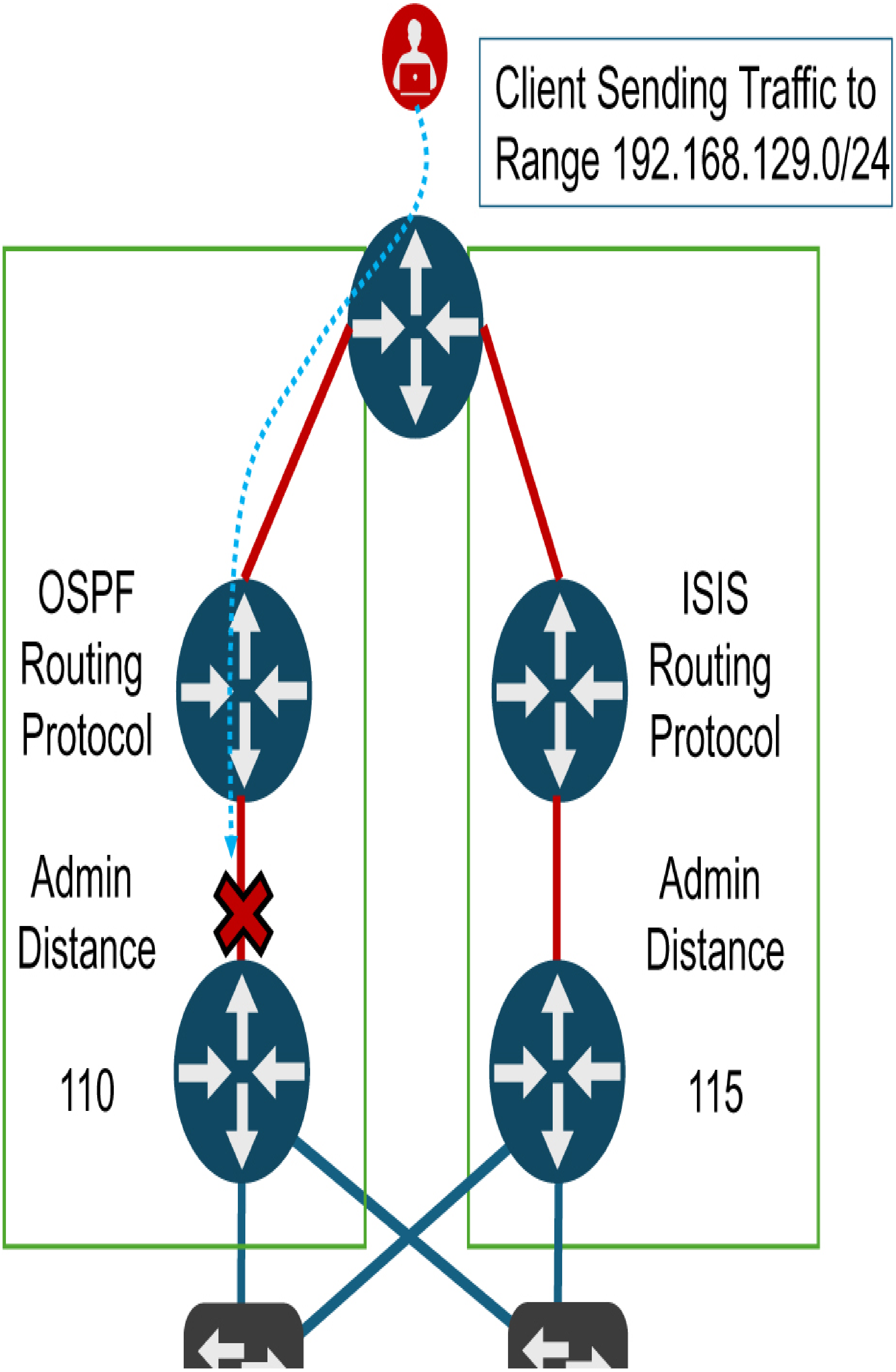


Network Protocol	Hello Interval	Hold Time	Administrative Distance
RIPv2	N/A	180 Seconds	120
EIGRP	5 Seconds	15 Seconds	Internal 90 External 170
EIGRP Low Bandwidth	60 Seconds	180 Seconds	Internal 90 External 170
OSPF	10 Seconds	40 Seconds (dead interval)	110
ISIS	10 Seconds	Hold time 30 Seconds	115
LISP	N/A	Cached up to 24 hours	250 (SDA)
OMP	60 Seconds	180 Seconds	250 (Viptella SD-WAN)
BGP	60 Seconds	180 Seconds	Internal 200 External 20

Figure 11-4 *IP Protocol Convergence Attributes*

Administrative distance to select a secondary route that may be available in another routing database is useful; however, the hold timers associated with the removal of a stale route from a routing protocol that may have a higher precedence can have an impact on the respective speed of convergence.

Figure 11-5 shows a situation in which a client that is attempting to reach network 192.168.129.0/24 is dropping traffic due to a disruption in the path. As the diagram shows, both OSPF and IS-IS protocols are propagating the network to the router that the client is connected to. The challenge, however, is that until the hold time expires on the failed path on the left, the second route with the higher administrative distance, learned via IS-IS, will not be installed.


Client Sending Traffic to Range 192.168.129.0/24
OSPF Routing Protocol
Admin Distance
110
ISIS Routing Protocol
Admin Distance
115

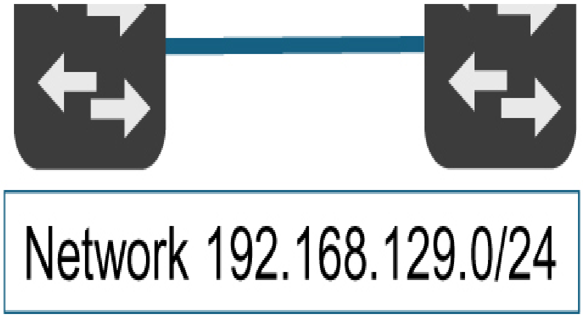


Figure 11-5 *Slow Convergence Attributed to Protocol Timers*

To expedite the speed at which a link is identified as being out of service, you can introduce timers and keepalives. Protocols such as bidirectional forwarding detection (BFD) can signal a loss state on a link or circuit prior to the network protocol, allowing for speedy convergence of network routing protocols, sending traffic over an alternative circuit or link. The aggressiveness that can be configured on BFD in terms of multiplier and echo intervals is heavily dependent on the hardware capabilities of the respective platform.

Example 11-4 is taken from a Catalyst 9300 Edge device, which was provisioned by Catalyst Center in an SD-Access network. Key points to note are the echo timer of 250 ms, the protocols that BFD is registered with (CEF and IS-IS), and the multiplier that controls the intervals related to failure detection timing.

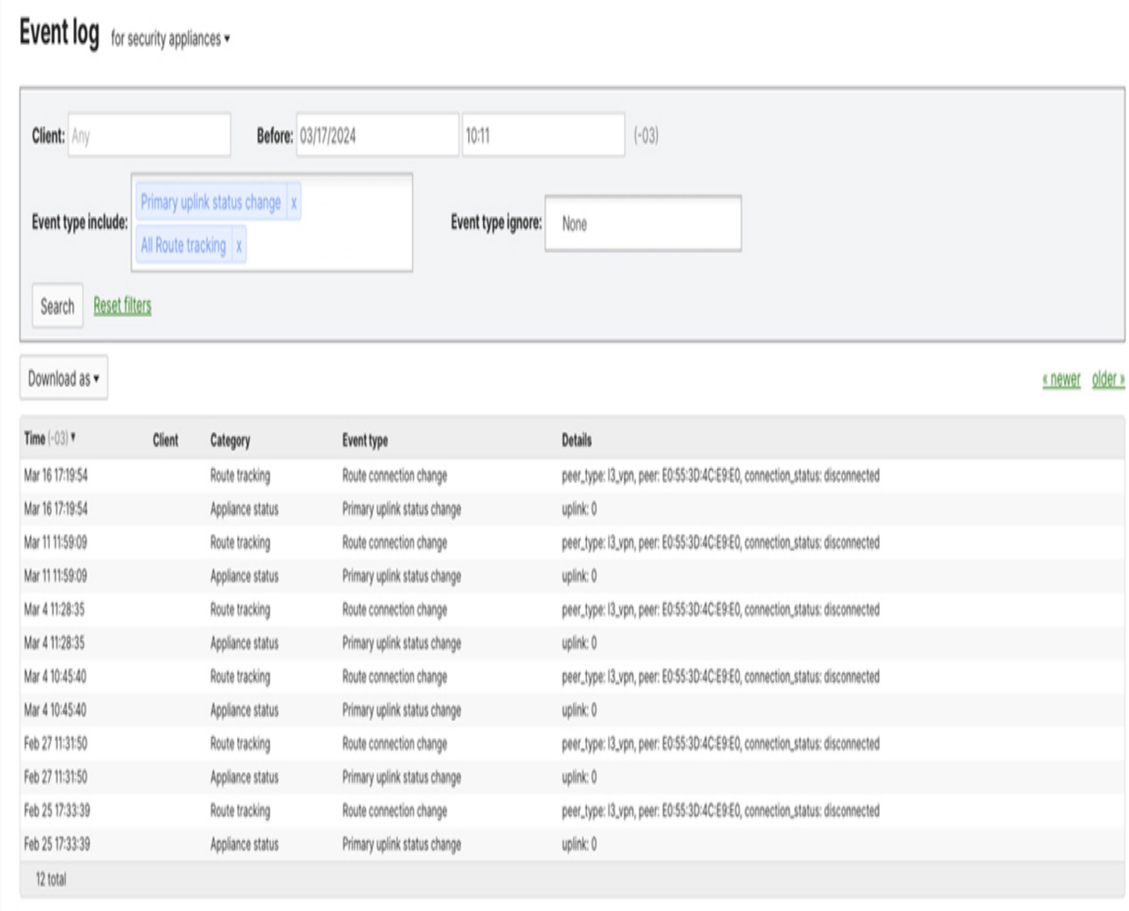
Example 11-4 *BFD Neighbor Overview*

```
Edge#show bfd neighbors  details
IPv4 Sessions
NeighAddr                                      LD/RD          RH/RS     St
172.16.10.68                                    4/1           Up        Up
Session state is UP and using echo function with 250 ms interval.
Session Host: Software
OurAddr: 172.16.10.69
Handle: 1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
```

```
Holddown (hits): 0(0), Hello (hits): 1000(945185)
Rx Count: 945202, Rx Interval (ms) min/max/avg: 1/1005/878 last:
Tx Count: 945187, Tx Interval (ms) min/max/avg: 1/1005/877 last:
Echo Rx Count: 3782434, Echo Rx Interval (ms) min/max/avg: 188/25
Echo Tx Count: 3782434, Echo Tx Interval (ms) min/max/avg: 189/25
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: ISIS CEF
Uptime: 1w2d
Last packet: Version: 1                  - Diagnostic: 0
             State bit: Up               - Demand bit: 0
             Poll bit: 0                 - Final bit: 0
             C bit: 0
             Multiplier: 3               - Length: 24
             My Discr.: 1                - Your Discr.: 4
             Min tx interval: 1000000    - Min rx interval: 10000
             Min Echo interval: 250000
```

The BFD protocol is also supported in conjunction with BGP, and it is strongly recommended to consider its use in production network environments on platforms that can support it. Using it can rapidly improve the speed at which convergence can take place, particularly in environments where a Layer 2 intermediate device such as a switch or media converter is in place, resulting in line protocols not automatically going down during a failure in the path, thus resulting in the routing protocol having to deal with the respective link failure detection. Tuning of the BFD parameters with the correct timers is heavily platform dependent. Given that some platforms handle different functions in hardware and have different scale limits, it is important to refer to the latest datasheets for the intended use to ensure that the ideal parameters are configured and that they will not cause any potential issues with the stability of the network.

In Meraki networks, network connectivity and history can be verified via the dashboard using the user interface or via the API, as shown in Figure 11-6.



Event log for security appliances ▾

Client: Any

Before: 03/17/2024 10:11 (-03)

Event type include: Primary uplink status change x, All Route tracking x

Event type ignore: None

Time (-03) ▾	Client	Category	Event type	Details
Mar 16 17:19:54		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Mar 16 17:19:54		Appliance status	Primary uplink status change	uplink: 0
Mar 11 11:59:09		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Mar 11 11:59:09		Appliance status	Primary uplink status change	uplink: 0
Mar 4 11:28:35		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Mar 4 11:28:35		Appliance status	Primary uplink status change	uplink: 0
Mar 4 10:45:40		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Mar 4 10:45:40		Appliance status	Primary uplink status change	uplink: 0
Feb 27 11:31:50		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Feb 27 11:31:50		Appliance status	Primary uplink status change	uplink: 0
Feb 25 17:33:39		Route tracking	Route connection change	peer_type: l3_vpn, peer: E0:55:3D:4C:E9:E0, connection_status: disconnected
Feb 25 17:33:39		Appliance status	Primary uplink status change	uplink: 0

12 total

Figure 11-6 *Tracking Routing and Uplink Changes in Meraki Networks*

In newer iterations of the Meraki MX security appliances that are often used in networks for WAN and uplink termination, the use of ThousandEyes observability monitoring is possible. It can provide a constant view and measurement of the application and network uplink health, providing clear and visible data on the speed at which convergence of the intermediate network takes place. Figure 11-7 shows service availability and health over time.

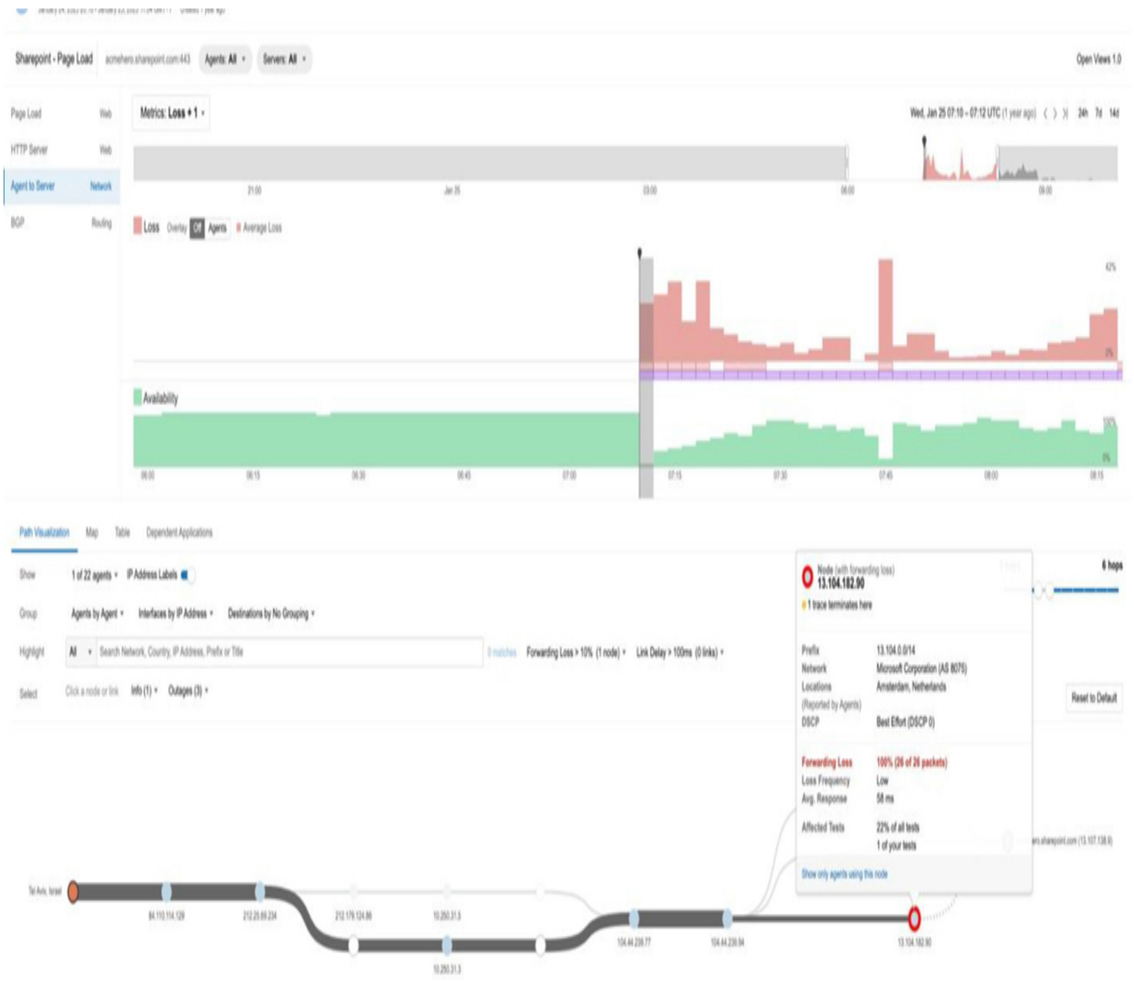


Figure 11-7 *ThousandEyes Dashboard: Showcasing Network Loss Toward a Remote Service*

To ensure a robust network topology that can support modern and secure applications and services, you should consider the following best practices:

- Leverage BFD on network links and routing protocols.
- Prune VLANs from trunks to limit the Layer 2 domain size.
- Reduce the number of routing protocols to avoid unnecessary complexity.
- Perform regular testing of redundant nodes to identify potential shortfalls in convergence.
- Evaluate the convergence needs based on new applications and services that require deployment.

Server-Side Verification

Deeper measurements to ascertain the impact of application latency and the ramifications of convergence are becoming more simplistic with the advancements in full stack observability, whereby service-to-service interactions can be measured in a granular way. Newer monitoring systems include the capability to provide end-to-end application measurements between nodes, servers, and containers.

While the focus of systems such as AppDynamics usually takes a service-side view, in IT networks, where the consumers of a service may exist within a location that is undergoing convergence, the knock-on effect and impact of an outage can also be measured, looking at an inverted load (drop in use) of key services.

This sort of scenario can be seen in Figure 11-8, where a significant load reduction occurs at 10:55 a.m. Because such monitorability considers high load and high application latency as negative factors, it is important to note that the opposite behavior would signal a usage impact, attributed to unreachable client-side consumers of the service, essentially meaning in many cases that a completely "green" dashboard may, in fact, be a negative consequence of network loss.

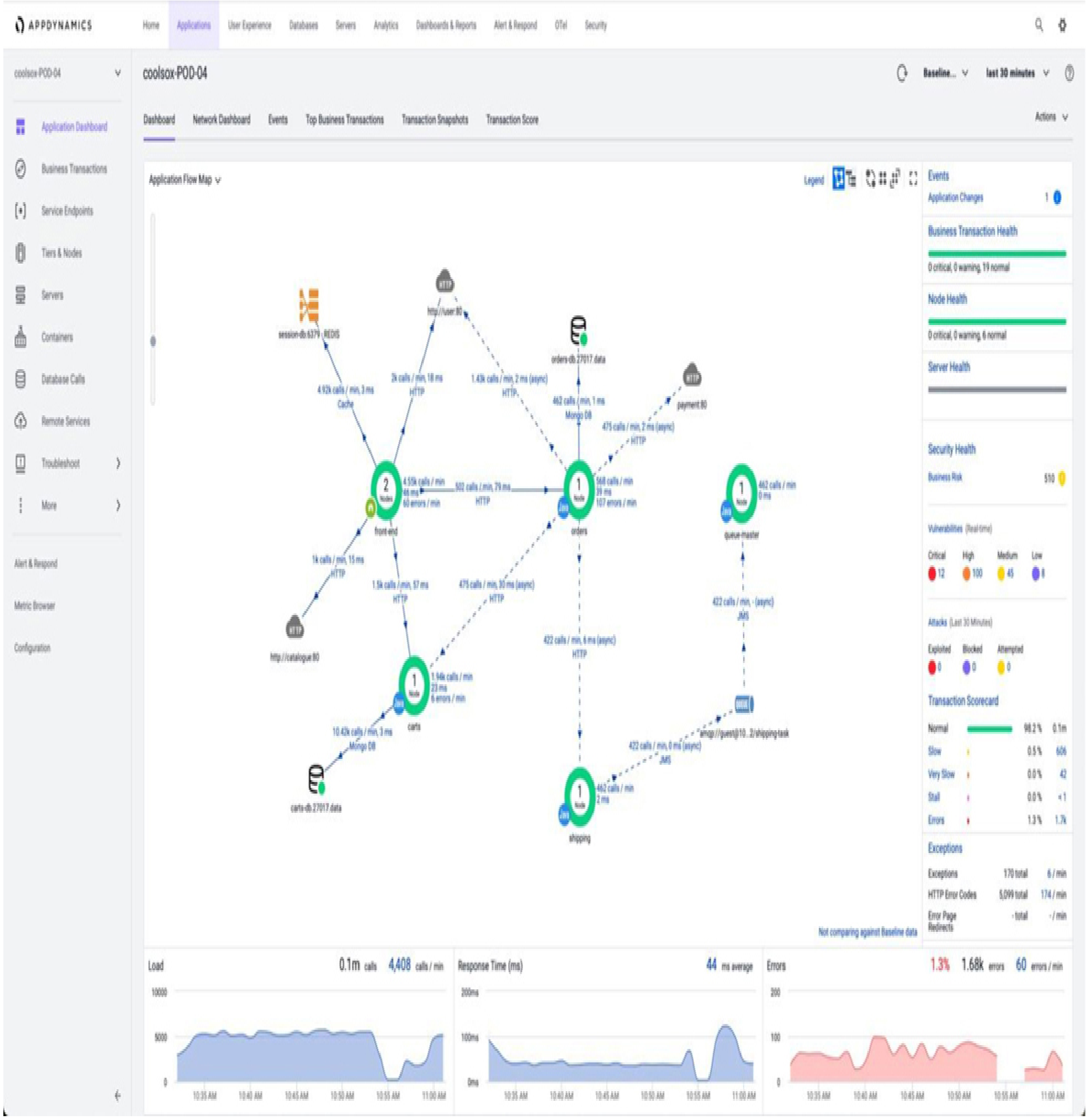


Figure 11-8 *Drop in Client Load in Application Side View*

Even without full stack observability platforms, you can still understand how applications may behave during convergence events by monitoring the network applications using traffic captures.

Capture points could be used for such assessments:

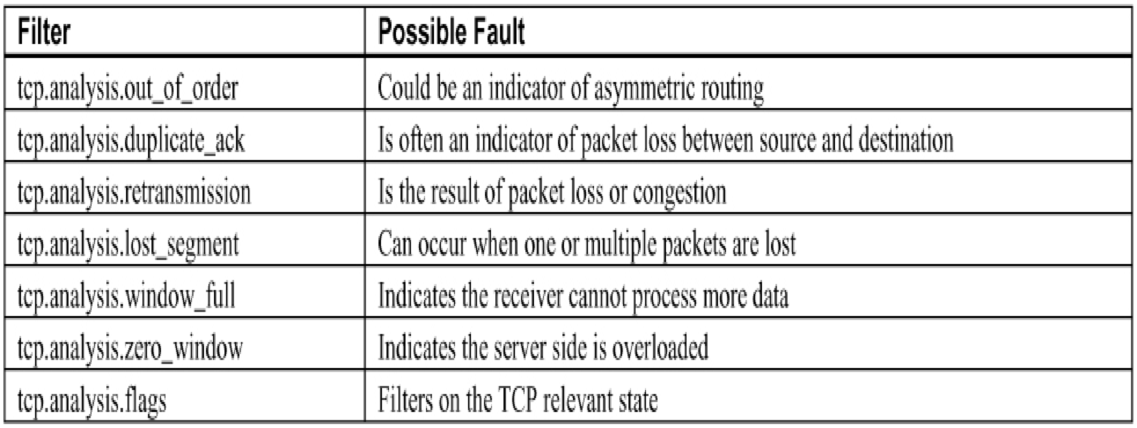
- A local system communicating with northbound applications affected by the convergence
- A SPAN session from a connecting switch to actively monitor VLANs or client traffic

- Monitor capture (may be limited by buffer size) on the connecting switch
- ERSPAN to a local docker container running Wireshark on supporting switches (e.g., a C9300 with an SDD120 module)
- A server-side capture from a connecting switch (may be difficult due to high throughput)
- A server-side capture on a system running an application instance (a respective pod in a microservices architecture)

Under certain circumstances, non-TCP—based applications are more difficult to understand and predict in terms of their failure behavior because much of the logic related to recovery and fault tolerance is written at the application layer. TCP-based applications tend to have a predictable logic that can be used as a means to understand the location of a particular fault in communications.

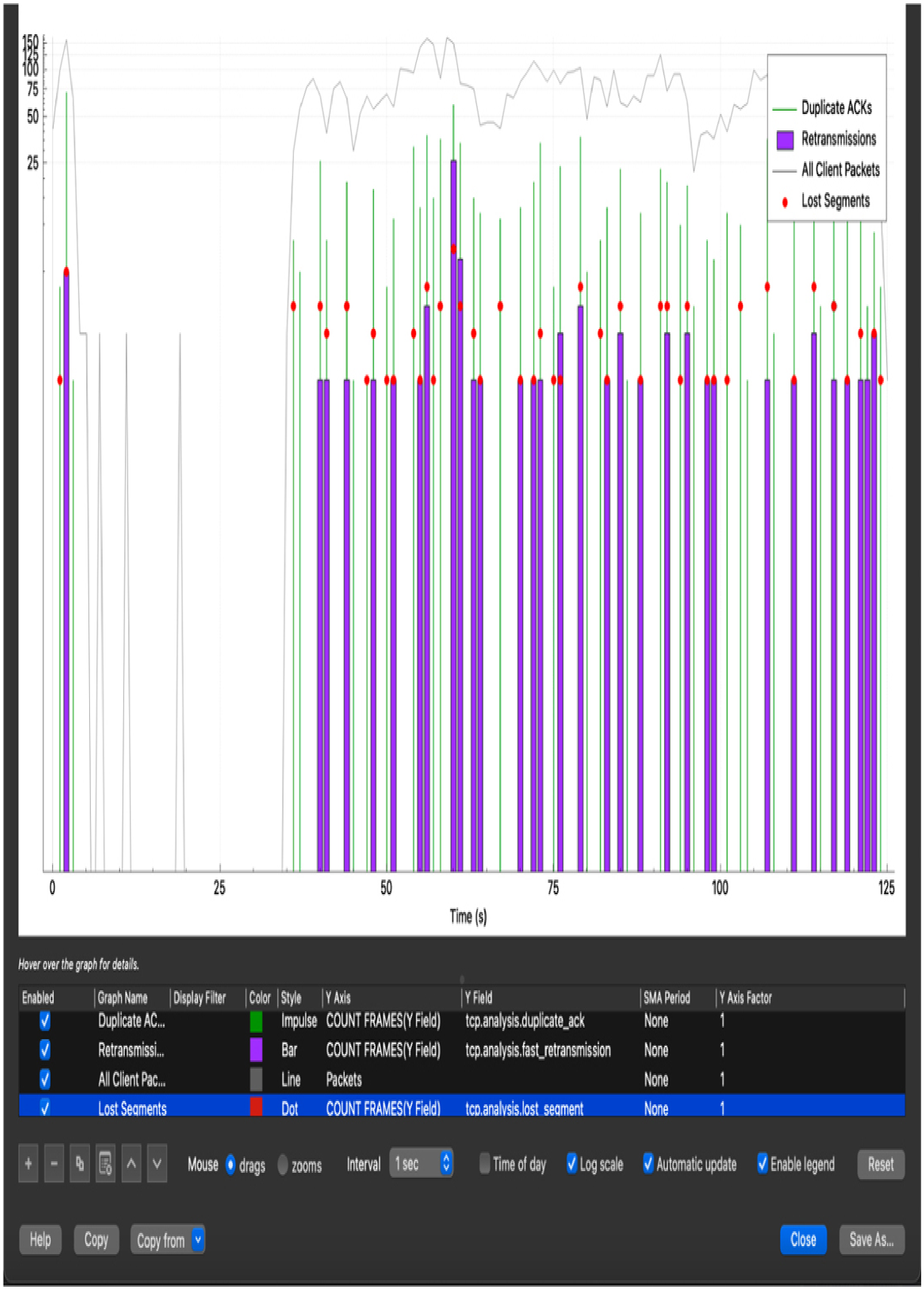
The Wireshark filters shown in Table 11-1 are useful in evaluating such problems.

Table 11.1 *Wireshark Filters*



Filter	Possible Fault
tcp.analysis.out_of_order	Could be an indicator of asymmetric routing
tcp.analysis.duplicate_ack	Is often an indicator of packet loss between source and destination
tcp.analysis.retransmission	Is the result of packet loss or congestion
tcp.analysis.lost_segment	Can occur when one or multiple packets are lost
tcp.analysis.window_full	Indicates the receiver cannot process more data
tcp.analysis.zero_window	Indicates the server side is overloaded
tcp.analysis.flags	Filters on the TCP relevant state

When you use these filters in I/O-based graphs in Wireshark, as depicted in Figure 11-9, a clear view of the application's impact becomes visible during convergence testing. Note, however, that it is very difficult to extrapolate good from bad when the concept of "good" is not properly understood. Different applications are programmed in different ways, often using different levels of network stack hygiene. Some applications are programmed to close with a TCP FIN, whereas others close with an RST (reset). Knowing this information during good times is helpful in being able to easily pick out erroneous behavior that is believed to be attributed to the network.



Hover over the graph for details.

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis Factor
☑	Duplicate AC...		🟩	Impulse	COUNT FRAMES(Y Field)	tcp.analysis.duplicate_ack	None	1
☑	Retransmissi...		🟪	Bar	COUNT FRAMES(Y Field)	tcp.analysis.fast_retransmission	None	1
☑	All Client Pac...		▨	Line	Packets		None	1
☑	Lost Segments		🟥	Dot	COUNT FRAMES(Y Field)	tcp.analysis.lost_segment	None	1

Mouse ● drags ○ zooms Interval 1 sec ☐ Time of day ☑ Log scale ☑ Automatic update ☑ Enable legend

Figure 11-9 *Wireshark I/O Graph Measuring TCP Health for Captured Application Communications*

Note

I/O graphing in Wireshark is a good way to benchmark and understand application health in the network—in particular, from a client-side perspective.

Convergence in Data Center Networks

Although there are a number of ways to perform convergence validations in DC architectures and designs that commonly deploy a spine-leaf construct, one of the execution options of convergence testing is to use the UNIX-based application iPerf.

In the following scenario, although we explore convergence within a spine leaf architecture, these same rules and principles could be applied to any array of other scenarios.

To determine the forwarding impact within a spine-leaf architecture data center design, you can use iPerf to qualify the time at which the forwarding within the data plane gets interrupted because of certain convergence events. Link transitions (up/down), device reboots, or the activation/deactivation of the maintenance mode of a spine or leaf node, also known as *graceful insertion and removal (GIR)*, will have a different forwarding impact on the data plane. With iPerf, you can determine the number of milliseconds that the forwarding will be impacted due to the convergence event that occurred. The difference between using OSPF or IS-IS is that an underlay routing protocol within the spine-leaf architecture or with or without bidirectional forwarding detection (BFD) can also be validated easily.

With two hosts connected to different leafs—either single-homed or connected with a virtual port channel (vPC) to different vPC domains—traffic can be transmitted between both hosts via the data center fabric.

In this example depicted in Figure 11-10, the iPerf client on one host sources UDP traffic to the iPerf server on the other host, which is listening on IP address 10.49.154.5, UDP port 5001. The iPerf client initiates 32 parallel UDP streams sourced from a random UDP port toward the destination on UDP port 5001. Each UDP datagram has a UDP payload of

1470 bytes (default value), which you can change by using the CLI command switch **-l** when executing the iPerf client command.

Each session receives an identification number—in this example, from 1 to 32. Within each session 1.000 packets per seconds (pps) are transmitted. This load is evenly distributed over the timeframe of 1 second for all sessions that are running concurrently. On average, the server sends 32 UDP datagrams, one for each session within 1 millisecond. The iPerf server is expected to receive 32.000 UDP datagrams per second, 1.000 for each stream. A summary that informs about the received amount of UDP datagrams per stream is printed every second.

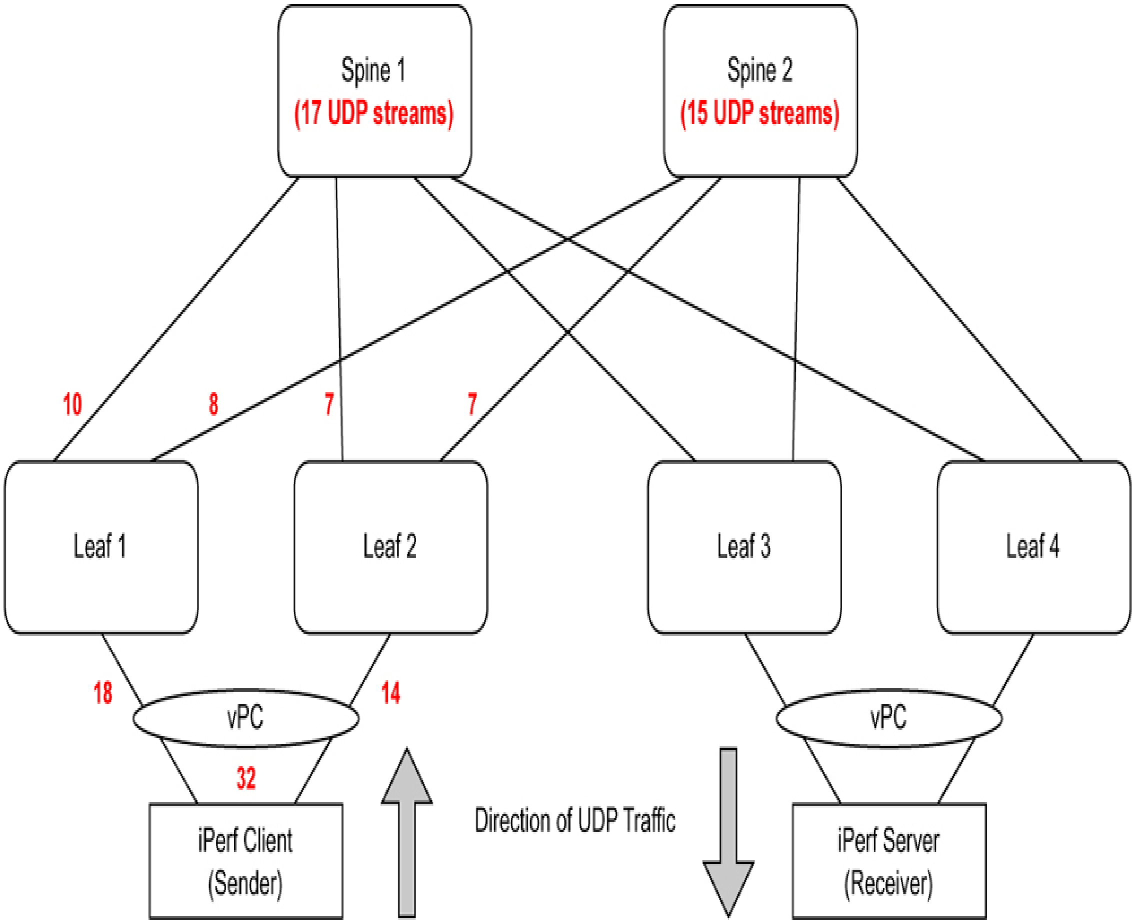


Figure 11-10 *Data Center Convergence Testing Topology*

Example 11-5 shows the output of the iPerf client with the IP address 10.49.154.4 (sending 32 UDP streams to the iPerf server with IP address 10.49.154.5 for 300 seconds). This is measured as a benchmarking exercise to validate a “good” state in the network without any network convergence activities taking place.

Example 11-5 *IPERF Convergence Testing Execution and Results*

```
user@iperf-client:~# iperf -u -c 10.49.154.5 -b 1000pps -t 300 -P
------------------------------------------------------------
Client connecting to 10.49.154.5, UDP port 5001
Sending 1470 byte datagrams, IPG target: 1000.00 us (kalman adjus
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.49.154.4 port 53396 connected with 10.49.154.5 por
[  4] local 10.49.154.4 port 51231 connected with 10.49.154.5 por
[  2] local 10.49.154.4 port 50108 connected with 10.49.154.5 por
[ 21] local 10.49.154.4 port 47554 connected with 10.49.154.5 por
[ 22] local 10.49.154.4 port 49866 connected with 10.49.154.5 por
[ 24] local 10.49.154.4 port 42741 connected with 10.49.154.5 por
[ 25] local 10.49.154.4 port 42206 connected with 10.49.154.5 por
[ 17] local 10.49.154.4 port 40796 connected with 10.49.154.5 por
[ 16] local 10.49.154.4 port 51693 connected with 10.49.154.5 por
[ 30] local 10.49.154.4 port 33559 connected with 10.49.154.5 por
[ 31] local 10.49.154.4 port 44815 connected with 10.49.154.5 por
[ 14] local 10.49.154.4 port 42533 connected with 10.49.154.5 por
[ 12] local 10.49.154.4 port 55266 connected with 10.49.154.5 por
[ 10] local 10.49.154.4 port 51489 connected with 10.49.154.5 por
[  8] local 10.49.154.4 port 54943 connected with 10.49.154.5 por
[ 27] local 10.49.154.4 port 48911 connected with 10.49.154.5 por
[ 19] local 10.49.154.4 port 38351 connected with 10.49.154.5 por
[  6] local 10.49.154.4 port 56793 connected with 10.49.154.5 por
[ 23] local 10.49.154.4 port 55713 connected with 10.49.154.5 por
[ 26] local 10.49.154.4 port 45995 connected with 10.49.154.5 por
[ 29] local 10.49.154.4 port 58414 connected with 10.49.154.5 por
[ 32] local 10.49.154.4 port 53209 connected with 10.49.154.5 por
[ 13] local 10.49.154.4 port 42865 connected with 10.49.154.5 por
[  9] local 10.49.154.4 port 36234 connected with 10.49.154.5 por
[  1] local 10.49.154.4 port 54389 connected with 10.49.154.5 por
[ 20] local 10.49.154.4 port 33236 connected with 10.49.154.5 por
[ 28] local 10.49.154.4 port 38783 connected with 10.49.154.5 por
[ 11] local 10.49.154.4 port 60463 connected with 10.49.154.5 por
[  5] local 10.49.154.4 port 44111 connected with 10.49.154.5 por
[ 15] local 10.49.154.4 port 36670 connected with 10.49.154.5 por
[  7] local 10.49.154.4 port 47527 connected with 10.49.154.5 por
[ 18] local 10.49.154.4 port 50680 connected with 10.49.154.5 por

iPerf server lisiten on IP address 10.49.154.5 (default UDP port

user@iperf-server:~# iperf -s -u -B 10.49.154.5 -i 1
------------------------------------------------------------
Server listening on UDP port 5001
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
...
[ ID] Interval             Transfer     Bandwidth        Jitter
[  1] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.011 ms
[  2] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.004 ms
[  3] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[  4] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[  5] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[  6] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.011 ms
[  7] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.004 ms
[  8] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.008 ms
[  9] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.007 ms
[ 10] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 11] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[ 12] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.004 ms
[ 13] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.007 ms
[ 14] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[ 15] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[ 16] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.015 ms
[ 17] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.007 ms
[ 18] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[ 19] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[ 20] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[ 21] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.007 ms
[ 22] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.035 ms
[ 23] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[ 24] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[ 25] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[ 26] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[ 27] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.015 ms
[ 28] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.013 ms
[ 29] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.011 ms
[ 30] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.011 ms
[ 31] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[ 32] 14.0000-15.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[SUM] 14.0000-15.0000 sec  44.9 MBytes   376 Mbits/sec   0/32000
```

The summary that is printed by the iPerf server shows the received amount of data, the calculated bandwidth, jitter, and lost/total datagrams. The summary does indicate the total for all of the 32 UDP streams. Based on the output showing that all streams have received 1.000 UDP datagrams, no UDP datagrams have been lost. The total amount of UDP datagrams received for the interval between second 14 to 15 is exactly 32.000. Depending on the system load/OS scheduling of the threads, this number can slightly vary by +/–10 packets. In case one stream was not able to send 1.000 datagrams, the total amount of UDP datagrams will indicate this. If we consider the influence of certain convergence events, the amount of data transmitted over time is not relevant in this case. What is important is the fact that we send 1.000 packets per seconds (pps) within each stream.

You need to ensure that the host and network will hash the 32 streams out of all available interfaces. For this to properly balance across available paths, you must take the L4 information into account. In the example provided here, the source and destination addresses are identical for all streams, as well as the UDP destination port. Only the source UDP port differentiates the 32 UDP streams. The idea is that the traffic between the iPerf client and iPerf server is hashed over all possible forwarding paths. This includes both vPC legs of each server; therefore, traffic will traverse both vPC leaf switches that the server is connected to. In an optimal scenario, 16 streams will traverse the left vPC leaf, and 16 streams will traverse the right vPC leaf.

As described previously, the ratio of distribution depends on the UDP source ports. With 32 streams, however, it is very unlikely that all the streams will go to only one vPC leaf. It can be assumed that there will be a distribution of the 32 streams between both leafs; for example, 18 will be transmitted toward the left vPC leaf, and 14 will traverse the right vPC leaf. The leaf will again distribute the streams between their uplinks. In this example, the left vPC leaf will utilize the first uplink to spine switch 1 with 10 streams and the uplink to spine 2 with 8 streams because it did receive 18 streams from the server via the vPC member port. The right vPC leaf will evenly distribute the 14 streams to both spines, with 7 UDP streams leaving each uplink because it did receive 14 UDP streams via the vPC member port that the iPerf source is connected to.

Now that you've observed a healthy network state, Example 11-6 shows a further scenario, where a link is disconnected, resulting in a change in the routing table. This changes the number of available equal-cost multipath links available and results in the current hashing of UDP streams being redistributed across the remaining available equal-cost links.

Example 11-6 shows the impact of this convergence event, whereby the link between Leaf 2 and Spine 2 in Figure 11-10 is disabled:

Example 11-6 *IPERF Network Actively Converging Based on an Event*

```
user@iperf-server:~# iperf -s -u -B 10.49.154.5 -i 1
------------------------------------------------------------
Server listening on UDP port 5001
UDP buffer size:  208 KByte (default)
```

```
------------------------------------------------------------
...
[ ID] Interval            Transfer     Bandwidth        Jitter
[  1] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.011 ms
[  2] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.003 ms
[  3] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[  4] 22.0000-23.0000 sec  1.15 MBytes  9.58 Mbits/sec   0.006 ms
[  5] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[  6] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.009 ms
[  7] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[  8] 22.0000-23.0000 sec  1.15 MBytes  9.59 Mbits/sec   0.006 ms
[  9] 22.0000-23.0000 sec  1.14 MBytes  9.56 Mbits/sec   0.012 ms
[ 10] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 11] 22.0000-23.0000 sec  1.15 MBytes  9.58 Mbits/sec   0.007 ms
[ 12] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.008 ms
[ 13] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.012 ms
[ 14] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.004 ms
[ 15] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[ 16] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 17] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 18] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 19] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 20] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 21] 22.0000-23.0000 sec  1.15 MBytes  9.58 Mbits/sec   0.007 ms
[ 22] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.008 ms
[ 23] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 24] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.004 ms
[ 25] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 26] 22.0000-23.0000 sec  1.16 MBytes  9.60 Mbits/sec   0.008 ms
[ 27] 22.0000-23.0000 sec  1.15 MBytes  9.58 Mbits/sec   0.004 ms
[ 28] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 29] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.005 ms
[ 30] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.006 ms
[ 31] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.010 ms
[ 32] 22.0000-23.0000 sec  1.40 MBytes  11.8 Mbits/sec   0.013 ms
[SUM] 22.0000-23.0000 sec  43.1 MBytes   361 Mbits/sec   1258/320
```

Example 11-6 shows that the convergence event (link shutdown) did happen in the iPerf testing interval 22 to 23. You can see that seven streams are impacted. Streams with the IDs 4, 8, 9, 11, 21, 26, and 27 show a loss of ~18 percent. Around 180 UDP datagrams packets of the expected 1.000 UDP datagrams have not been received by the iPerf server for each of the impacted UDP streams. The forwarding impact of the impacted link is around 180 milliseconds. For this amount of time, forwarding over this link is not possible. For the iPerf example, this results in a loss of 1.258 UDP datagrams out of the 32.000 total UDP datagrams (a total loss of UDP datagrams of the seven impacted UDP streams). If the iPerf sender is started again, the test can be repeated. Because the source UDP ports will change each time, you will very likely see a different number of streams with different IDs impacted.

The important value, however, is the loss of UDP datagrams per stream that should be approximately the same number for each stream. As you send 1.000 packets per second (pps) per UDP stream, evenly distributed over 1 second, you can interpret one missing UDP datagram with 1 millisecond of forwarding impact. This value will be fairly consistent when testing the impact of the same convergence event (link down).

Further tests that would be possible in such a topology could be

- The impact when spine 1 goes down (loss of power)
- The impact when spine 1 comes back with all links enabled
- The impact if it comes back when it boots with maintenance mode enabled when IS-IS initially has the overload bit enabled

Example 11-6 depicts one of many convergence scenarios that could be applied to a data center deployment. To ensure that such a failure or other failures that may occur can limit the time to converge as rapidly as possible, you should consider the following best practices:

- Summarize routes where possible when within a Layer 3 Fabric.
- Apply load to a data plane only into the respective fabric that requires it.
- Identify scale considerations based on the weakest platform (based on datasheet specifications).
- Reduce routes within the IGP; this practice is highly recommended.
- Avoid adjusting default timers within the IGPs; instead, focus on using BFD.
- Leverage BFD echo wherever possible and supported.
- In select multi-vendor scenarios utilizing EVPN, you might need to increase numbers for prefix propagation to accommodate different EVPN route types.

Convergence in Software-Defined Architectures

In software-defined architectures, a mix of protocols often is at play to achieve more advanced network functioning. In the context of software-defined access, the LISP protocol is used to provide more dynamic options in terms of traffic steering and mobility. Other software-defined deployments such as EVPN or SD-WAN may exercise the use of other protocols such as OMP or multi-protocol BGP to achieve the required function. Under certain circumstances, there are also multiple points in the network where redistribution may take place between different network protocols. These different points of entry for routing information need to be considered as part of the end-to-end network convergence activities that must take place.

The two scenarios showcased in Figure 11-11 detail how potential convergence situations may look in the case of varied network architectures. The first network design uses SD-Access in headquarter and branch locations with the Cisco software-defined WAN as the transit method connecting the disparate locations.

When looking at the end-to-end network protocols involved, you can observe the following in Figure 11-11.

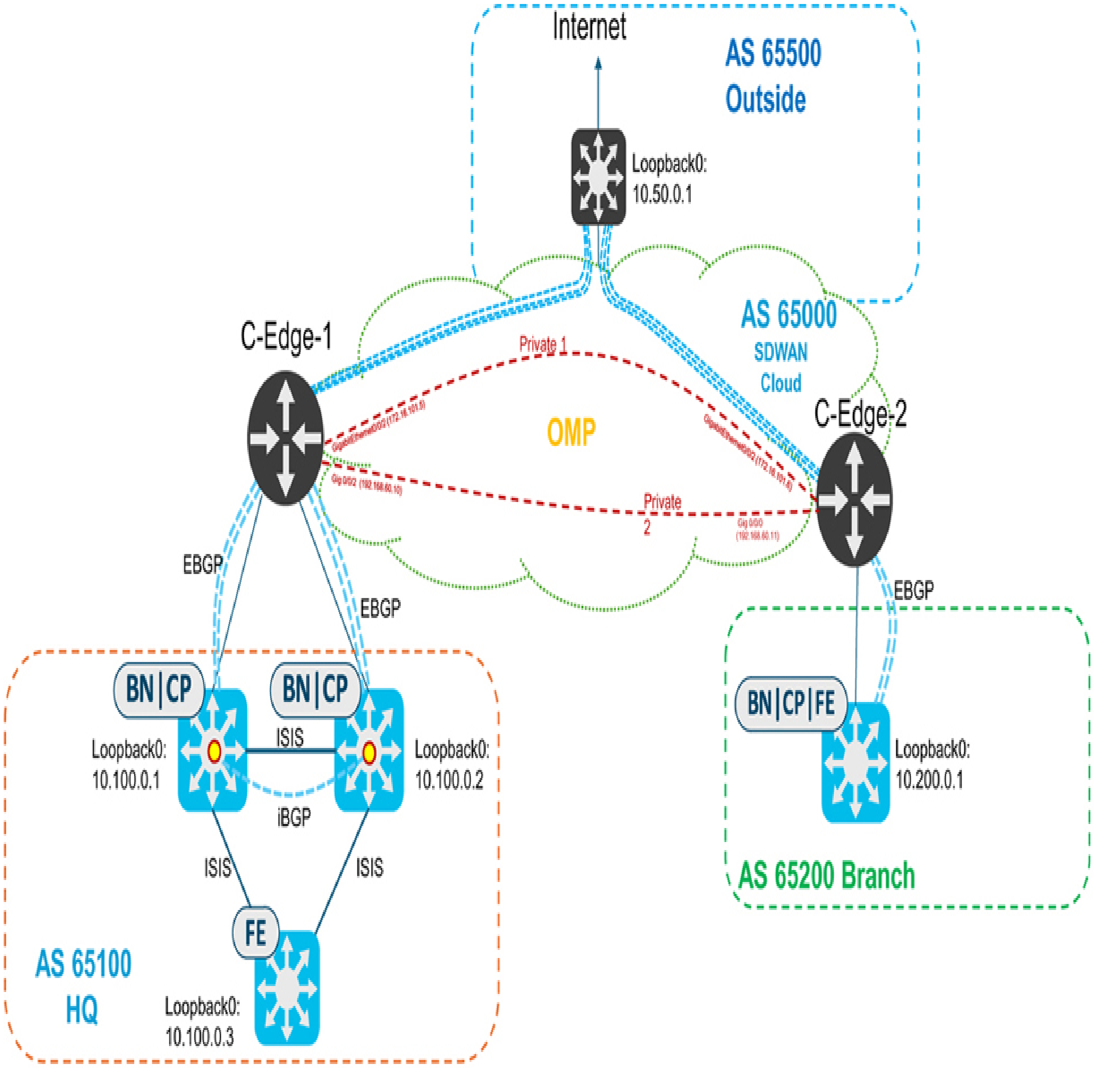


Figure 11-11 *Underlay Protocol Interaction Between Multiple Software-Defined Architectures*

In the network underlay, the use of the following protocols is in place:

Underlay: IS-IS → BGP → OMP → BGP → IS-IS

LISP also used in select scenarios (AP Pool and Extended Node)

Viable network convergence depends not only on the respective timers, interface, link failure detection, and protocol convergence times that are relevant to the reboot of a node, but also the speed of the redistribution updates and changes in and out of the respective IP routing protocols. Due to SD-Access and SD-WAN architectures natively using BFD, you should assume that the convergence is relatively rapid within their domains. In scenarios where there is a handoff between domains, such as the juncture point between SD-Access and SD-WAN, however, by default, BFD may not be enabled and under certain circumstances can represent a point where convergence may be affected by detection of a failure.

Verifying and validating such scenarios when bringing up a new network and on a regular basis over time can ensure that the network will maintain service during outage scenarios. You should measure the following during convergence testing:

- Impact (duration of convergence) of the loss/recovery of a link
- Impact (duration of convergence) of the loss/recovery of a network node
- Impact (duration of convergence) of multifailure scenarios
- Impact during software upgrade scenarios
- Impact of per protocol clearing that is often performed during troubleshooting (and convergence time associated)

Underlay networks are often the foundation for more advanced topologies; such networks are often the first points to converge.

In terms of convergence in the domain of security protocols, there are numerous protocols that come into play in modern networks. For instance, although MACsec—which is deployed to encrypt traffic that traverses a network link or connection—is not activated by default, in SD-Access, it is an underlay protocol that can be used for hardening network architectures. Its respective configuration and bring up time can also be considered an attribute in terms of convergence of the network security state.

Deploying TrustSec enforcement on the underlay topology is often avoided, or static SGT bindings and policies are applied. The main reason for avoiding TrustSec enforcement on the underlay infrastructure is so that systems that need to communicate with key components and services, such as the ISE Radius architecture, and establish routing protocol adjacencies can do so unimpeded.

The overlay network, as depicted in Figure 11-12, introduces some extra protocol considerations to the mix. First, LISP is used as a means to ensure that there is endpoint mobility and is a key contributor in maintaining roaming of prefixes.

In contrast to the underlay network, which consists of protocol communication paths that often represent the physical links and infrastructure, the overlay network provides a more logical representation of control plane constructs that have an effect on the forwarding of traffic.

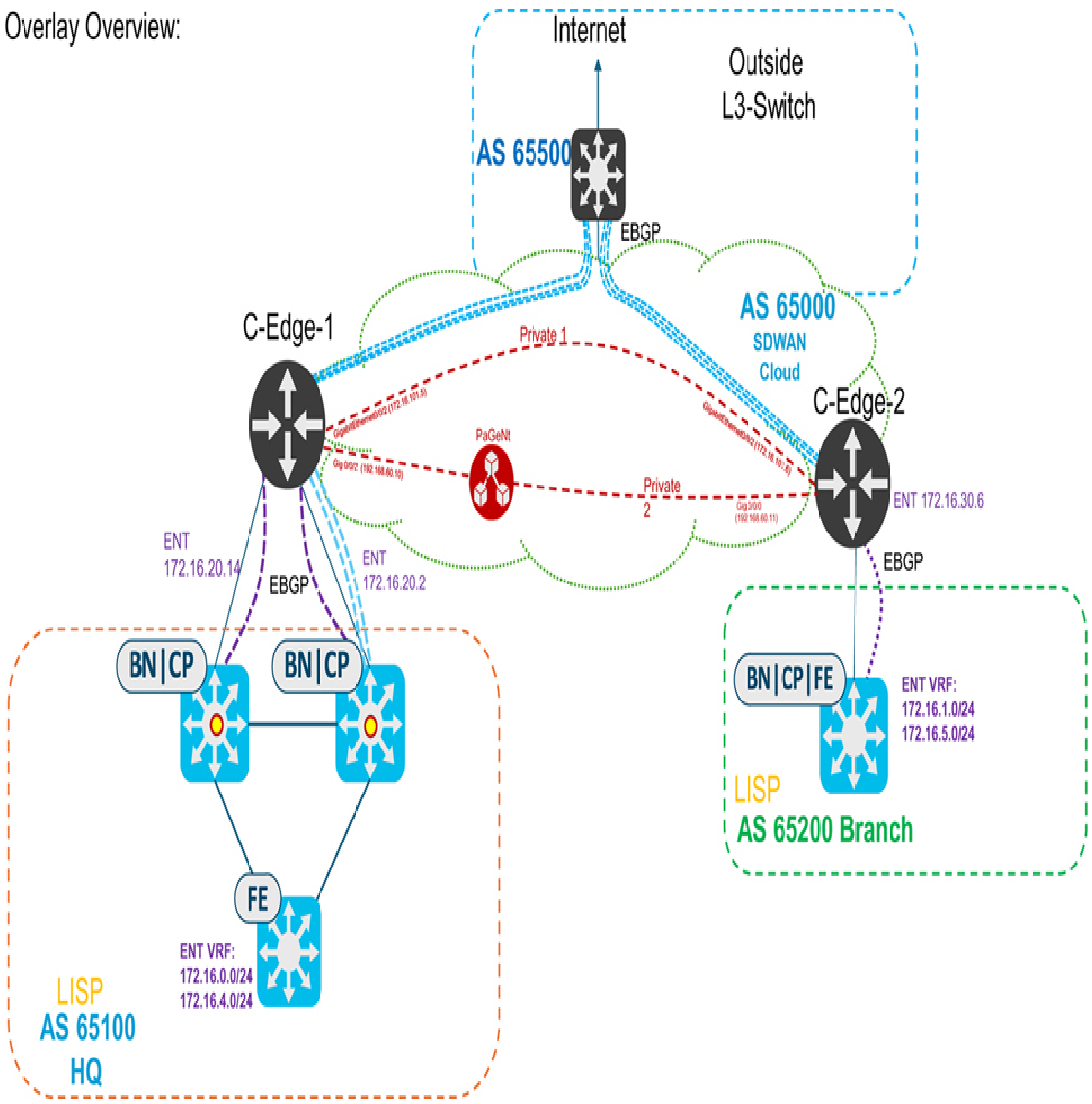


Figure 11-12 *Overlay Protocol Interaction Between Multiple Software-Defined Architectures*

In the network overlay, the following protocols are in place:

Overlay: LISP → BGP → OMP → BGP → LISP

When we look at traffic forwarding in an overlay network architecture, traffic for technologies such as EVPN and SD-Access often traverses the network in VXLAN tunnels.

SD-Access (IP Transit) in an overlay network represents a deployment scenario in which there is a handoff from a site to a *fusion layer*, which may be firewalls, CPE routers from a service provider, or a dedicated demarcation set of switches to connect domains. In networks that are built around the use of macrosegmentation and the maintenance of VRFs at the fusion layer (via VRF Lite), injection into an MPLS VPNv4 routing table is performed to preserve the macro segmentation separation across the network. For microsegmentation to be maintained, Cisco TrustSec (CTS) inline tagging up to the fusion layer is commonly used to ensure that tags are maintained. Convergence scenarios from IP Transit can be slow if nonrecommended physical topology constructs are used, and protocols such as BFD are not used in conjunction with the network handoff.

SD-Access (SDA Transit) in overlay architectures simplifies software-defined network deployments and the maintenance of both macro- and microsegmentation, by including the VN (VRF) and SGT information within the VXLAN header. This allows site constructs to be instantiated where only global routing table access is required from site to site, with the overlay information being maintained from a control plane protocol perspective via the LISP routing protocol. From a convergence perspective in an SD-Access Transit architecture, the publisher/subscriber model is used, with communications from site borders taking place to SD-Access Transit control planes that maintain the relevant mappings of prefix information to the borders that act as their exit points. These devices are often comparable to BGP route reflectors that are used in other domains.

Using configurations such as LISP affinity, which allows for ingress routing path prioritization, can provide a means for site-specific forwarding. In terms of resilience, through the way that affinity prioritization can be performed, primary and secondary and tertiary exit points for the default route could be considered, for example, and as such, Figure 11-13 shows a scenario in which measuring convergence via these disparate paths could be considered.

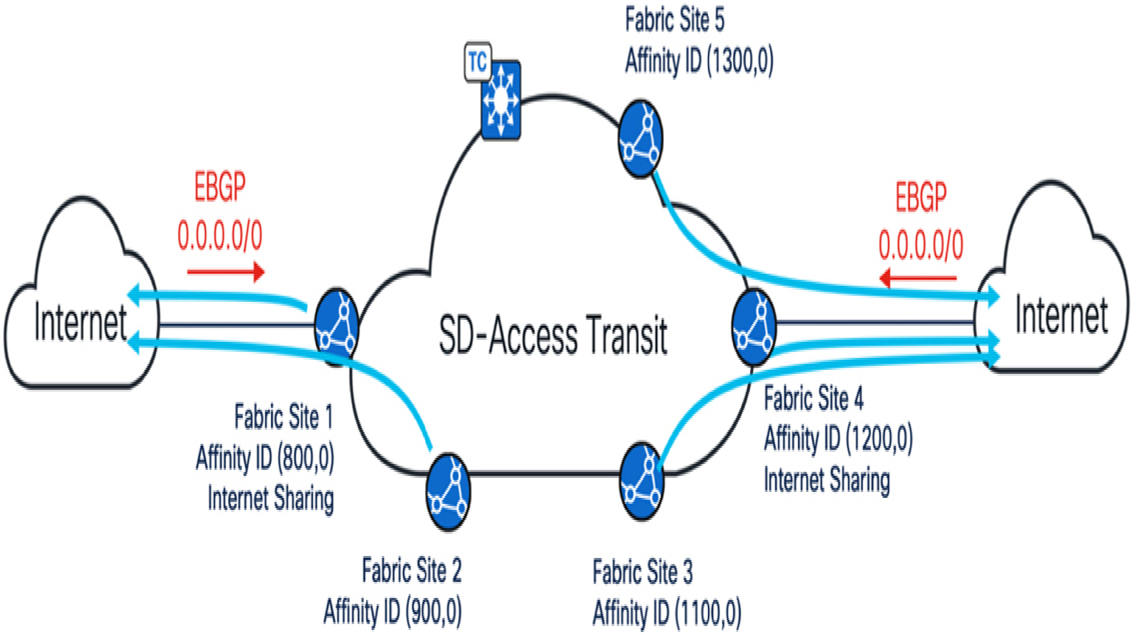


Figure 11-13 *Utilizing LISP Affinity to Perform Selective Traffic Engineering*

Under certain circumstances, SD-Access Transit affinity can also be considered part of a survivability design, when considering upgrade scenarios within a network that requires high resiliency.

From a security preservation perspective, for data plane traffic, through the use of VXLAN, maintenance of tags is achieved inline as part of the end-to-end client data encapsulation.

EVPN in a campus network can be closely compared to SD-Access, in the context of it being an overlay-based architecture, allowing for the preservation of key segmentation data within the VXLAN header. Convergence validation for this architecture mirrors that of SD-Access; however, BGP does not award the same flexibility that is present within the LISP protocol for certain selective forwarding scenarios. Therefore, outcomes that are available through configurations such as LISP affinity, which allows for a level of selective traffic engineering, are not present in BGP and may result in suboptimal routing configurations that would prompt further configuration overhead and complexity, such as VRF table separation to achieve desired traffic forwarding patterns.

Methodologies of Convergence Testing

Prefix length, admin distance, metric and equal-cost multipath, and TrustSec binding table numbers of dynamic SGACLs are all key factors in identifying the forwarding path and security configurations being appropriately applied in networks that are in convergence. To properly validate and test convergence, these factors require the right level of consideration, because they will manifest themselves in the outputs that are seen during a network impact.

From a routing perspective, Cisco Express Forwarding takes the key information into account when selecting the hash-based path to use to reach a particular exit point for traffic.

Example 11-7 shows that within the same IP subnet (172.16.1.0/24) different equal-cost exit points will be used, based on the hashing algorithm. In some older platforms, this algorithm is as basic as a three-tuple calculation, such as source, destination, and lowest loopback. Newer platforms and newer software tend to take a more advanced approach to hashing, resulting in a more even spread.

Example 11-7 *Forwarding Calculation Based on IP Forwarding Information Base*

```
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.4
172.16.0.4 -> 172.16.1.4 =>IP adj out of GigabitEthernet1/0/1, ad
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.5
172.16.0.4 -> 172.16.1.5 =>IP adj out of GigabitEthernet1/0/1, ad
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.6
172.16.0.4 -> 172.16.1.6 =>IP adj out of GigabitEthernet1/0/23, a
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.7
172.16.0.4 -> 172.16.1.7 =>IP adj out of GigabitEthernet1/0/1, ad
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.8
172.16.0.4 -> 172.16.1.8 =>IP adj out of GigabitEthernet1/0/23, a
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.9
172.16.0.4 -> 172.16.1.9 =>IP adj out of GigabitEthernet1/0/1, ad
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.10
```

```
172.16.0.4 -> 172.16.1.10 =>IP adj out of GigabitEthernet1/0/1, a
Edge#
```

On IOS XE platforms, it is also possible to identify how well distributed the hashing algorithm calculations are based on the source-to-destination range, as shown in Example 11-8. This capability is particularly useful when you’re looking at hashing algorithms working with the forwarding of heavy traffic loads.

Example 11-8 *Prefix Hash Distribution Based on FIB Destination Prefix Range*

```
Edge#show ip cef vrf ENT exact-route 172.16.0.4 172.16.1.10 dest-
Calculating OCE counts...

OCE counts (2 oces used for 245 sessions)
 OCE 0 : 115 sessions
 IP adj out of GigabitEthernet1/0/23, addr 172.16.10.20
 OCE 1 : 130 sessions
 IP adj out of GigabitEthernet1/0/1, addr 172.16.10.10

15 sessions (6.12%) not ideally load shared
Edge#
```

Figure 11-14 depicts a simple example of equal-cost load balancing. In this particular scenario, the cumulative bandwidth transmitted exceeds the available bandwidth of a single link, requiring the load to be properly balanced to prevent traffic from being dropped.

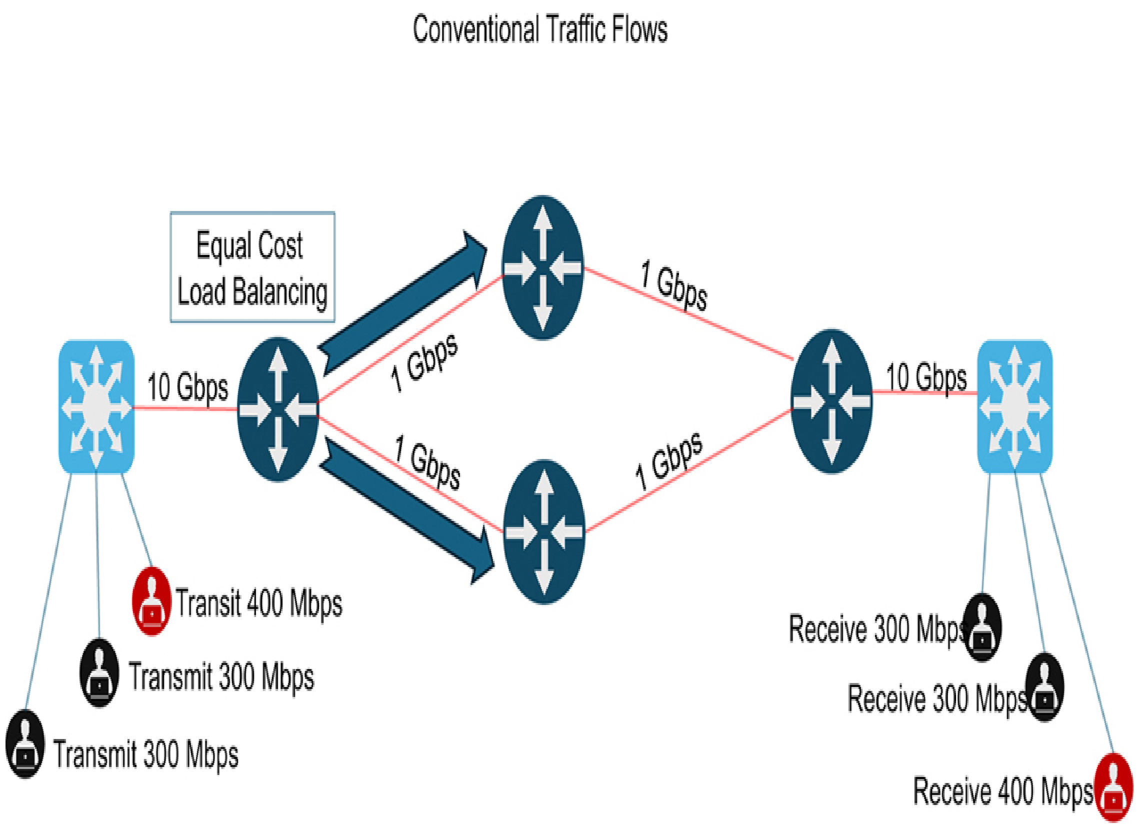


Figure 11-14 *Equal-Cost Multipath Flows of Traffic over a Network*

In addition to looking in Cisco Express Forwarding for how the forwarding path is being represented, topology databases and per protocol tables can provide a helpful view to help you understand the per protocol justifications for why a particular routing path is being installed. Example 11-9 shows how the LISP protocol has cached such an entry.

Example 11-9 *Multiple Remote Exit Points to Reach a Network via LISP*

```
Edge#show lisp instance-id 4099 ipv4 map-cache 172.16.1.0/24
LISP IPv4 Mapping Cache for LISP 0 EID-table vrf ENT (IID 4099),

172.16.1.0/24, uptime: 1d07h, expires: never, via pub-sub, comple
  Sources: pub-sub
  State: complete, last modified: 1d02h, map-source: local
  Exempt, Packets out: 469(268543 bytes), counters are not accura
  Configured as EID address space
  Locator       Uptime    State  Pri/Wgt     Encap-IID
  10.100.0.2    1d07h     up      10/10        -
    Last up-down state change:         1d02h, state change count:
    Last route reachability change:    1d02h, state change count:
    Last priority / weight change:     never/never
    RLOC-probing loc-status algorithm:
      Last RLOC-probe sent:            never
  172.16.10.65  1d02h     up      10/10        -
    Last up-down state change:         1d02h, state change count:
    Last route reachability change:    1d02h, state change count:
    Last priority / weight change:     never/never
    RLOC-probing loc-status algorithm:
      Last RLOC-probe sent:            never
Edge#
```

One challenge and risk with software-defined overlay protocol use, such as VXLAN, is that traffic from source to destination is encapsulated from loopback-to-loopback address on the source site to the destination site, or source switch to destination switch. From a logical perspective, this approach is feasible and configurable; however, in scenarios where the intermediate network may be using substandard hashing algorithms, such as two-tuple hashing, there could be situations where all the traffic traverses a single path and, given the bandwidth transmitted, would result in drops (see Figure 11-15).

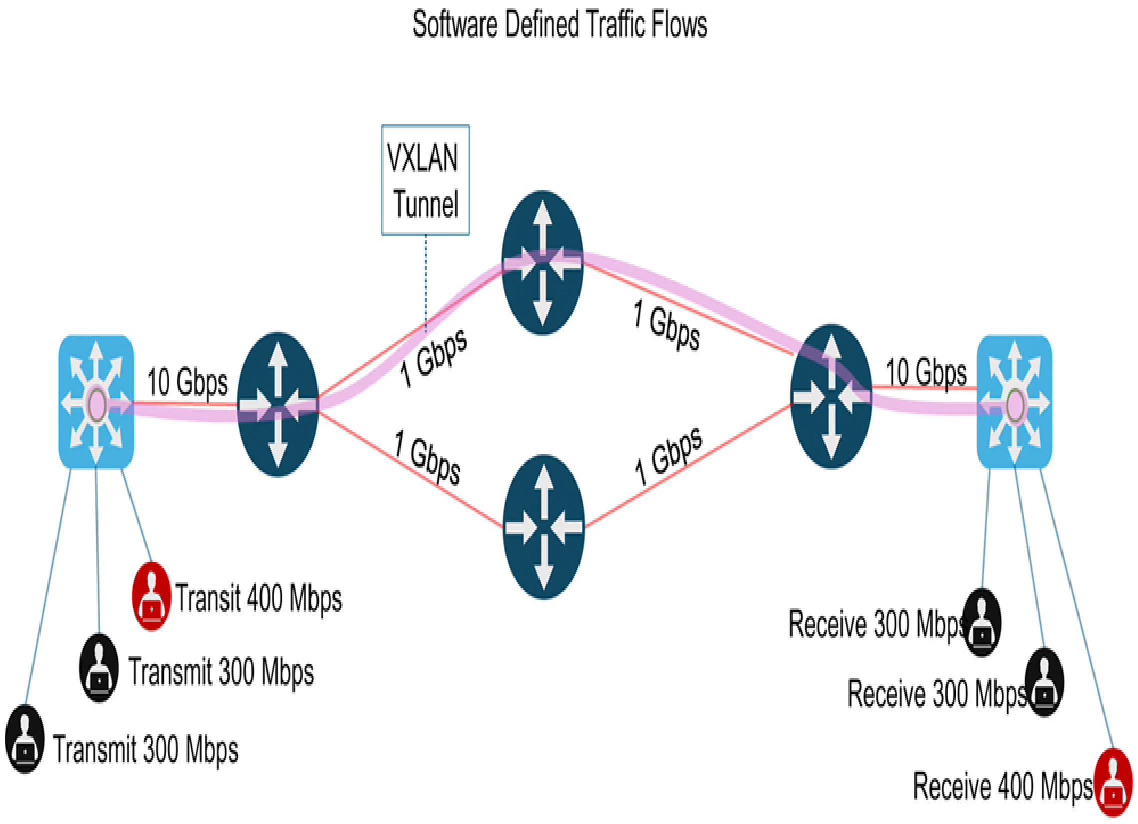


Figure 11-15 *Suboptimal Overlay Traffic Flows*

To address this issue, you can do a number of things. You can configure the hashing algorithms that are in use to increase the number of tuples in their hashing algorithm—for instance, moving from four tuples to seven tuples on supporting platforms. Example 11-10 shows a variety of options that are available to change the default hashing configuration.

Example 11-10 *Layer 3 Configurable Hashing Options*

```
Border(config)#ip cef load-sharing algorithm ?
  include-ports  Algorithm that includes layer 4 ports
  original       Original algorithm
  src-only       Algorithm that uses Src Addr only
  tunnel         Algorithm for use in tunnel only environments
  universal      Algorithm for use in most environments

Border(config)#
```

Simulating Routed Convergence Scenarios

As the demands and needs on modern networks increase, it is no longer just service providers and vendors that need to use more savvy and reliable methods to validate the integrity of modern architectures. To achieve this goal, a much heavier focus is being placed on systems that can provide a more accurate benchmark than a simple ping test. As we learned previously, using it as a viable measure is often challenging, especially given the way that hardware forwarding algorithms such as Cisco Express Forwarding operate in the context of the hashing of equal-cost forwarding paths.

Deploying traffic generators within test architectures introduces a few naming conventions that may be unfamiliar to some network operators. This naming refers to device under test (DUT). Conventionally, a traffic generator was deployed in a manner to achieve throughput validation for a specific device (the device under test). In modern networks, the device under test is no longer limited to a single device, but rather numerous devices, making the DUT the transit network between ports of the traffic generator.

When we look at options for how a traffic generator may be used within a network architecture, typically two options require initial consideration: Should the deployment be configured with a stateful or stateless traffic generation profile?

What Is Stateful Mode?

Stateful traffic generation refers to the creation of traffic streams that maintain a level of “state” and intelligence, when being transmitted. For instance, TCP initially performs a three-way handshake prior to the transmission of network payloads and data. As part of this handshake, intermediate devices, such as firewalls and load balancers, often maintain stateful tables to ensure that specific sessions are permitted, or balanced, toward the same target server-side resources. The purpose is to avoid that TCP traffic from a client session that was instantiated to a specific back-end server or an application instance is sent to an unrelated server that is not aware of the stateful TCP session. Through intermediate devices maintaining these stateful mappings, they therefore avoid sending packets that are not related to the corresponding client to server TCP session to a system that doesn't have knowledge of the respective session which was established.

Figure 11-16 shows a sample traffic generator topology for stateful traffic.

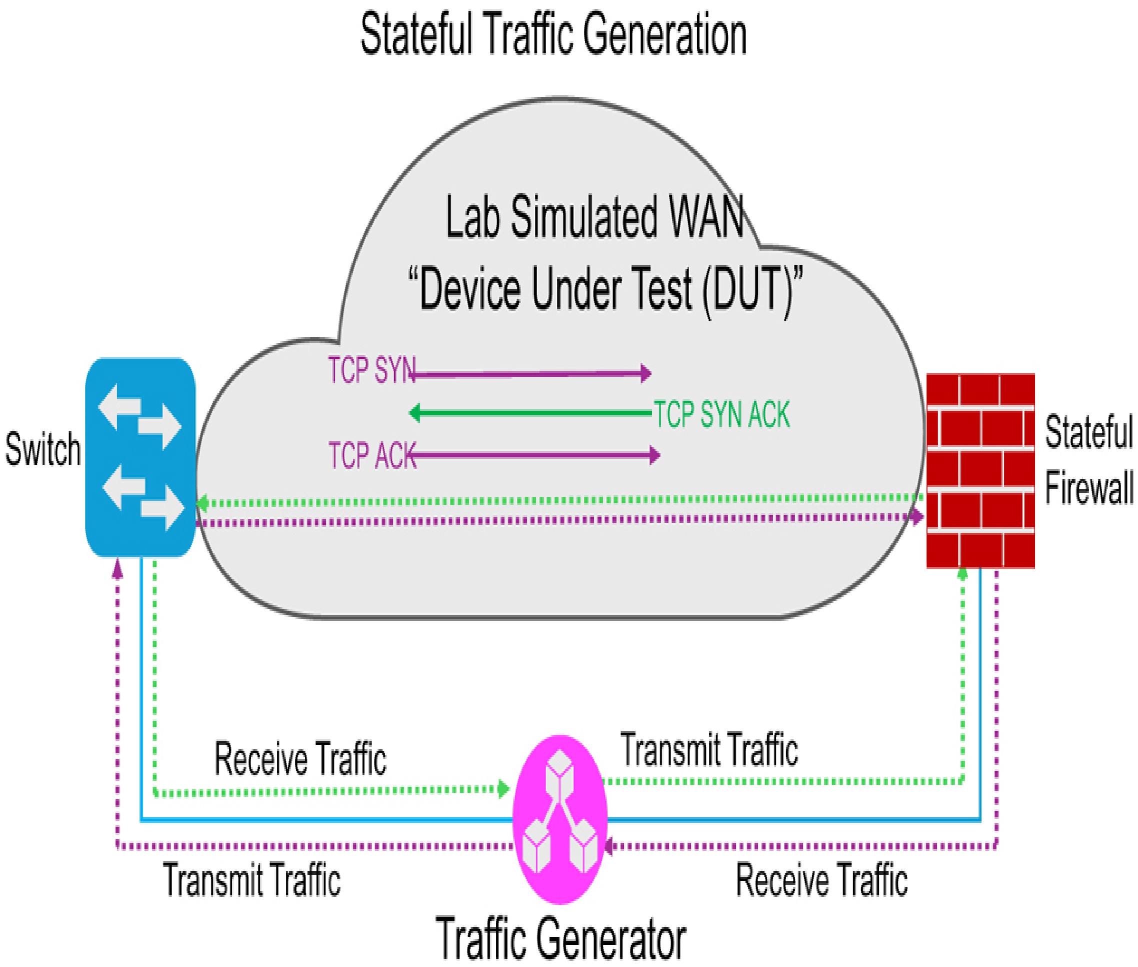


Figure 11-16 *Stateful Traffic Generator Setup*

For testing convergence of traffic through stateful firewalls and load balancers, stateful traffic generation mode is usually mandatory. As such, traffic profiles that meet this need are used.

What Is Stateless Mode?

Stateless traffic generation has the opposite characteristic of traffic generation to stateful traffic generation. No TCP/UDP application layer session is being maintained and generated. Likewise, no handshakes and no session information encodings that would allow a firewall or load balancer to properly ensure "sticky" mappings of traffic profiles are maintained.

While this way of operating may sound like a negative thing, it actually isn't, not if the intention for the tests is to validate convergence or throughput, for example, where the only concern is the number of packets that arrived or were dropped. Also, this way of operating isn't bad if we consider the duration of the outage scenario when traffic is sent over a particular link or set of links with a disrupted transit path. Figure 11-17 shows the differences in how such a topology could look when using stateless mode for traffic generation.

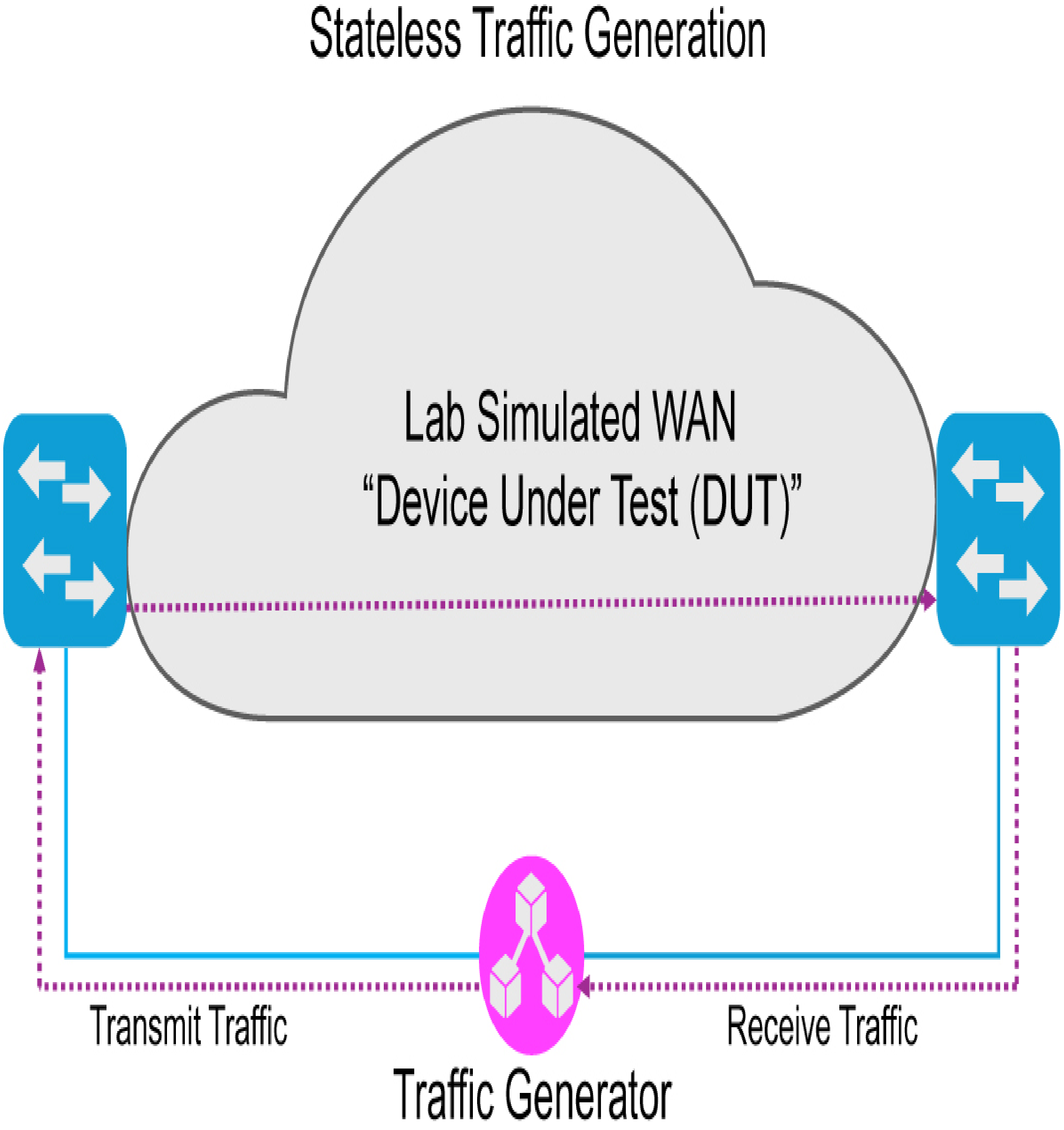


Figure 11-17 *Example Topology Used for Stateless Traffic Generation*

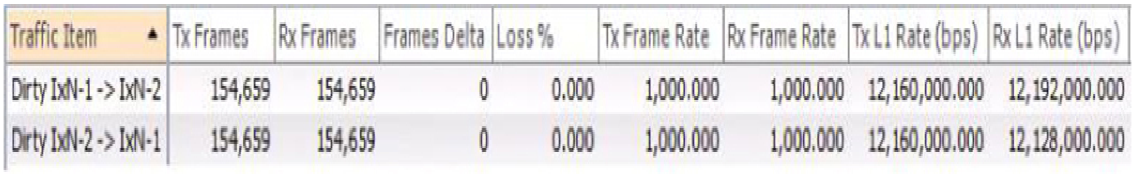
In most scenarios where convergence is a key consideration for a DevOps engineer, stateless is the preferred method used to verify the convergence state.

At the time of writing, three traffic generators were seen in the field. Each has its own set of capabilities, advantages, and or disadvantages when being used:

- IXIA from Keysight
- Spirent from Spirent Communications
- TRex (open source; initially developed by Cisco Engineers)

When you’re testing, it is important to benchmark a healthy and working state prior to the execution of any convergence testing.

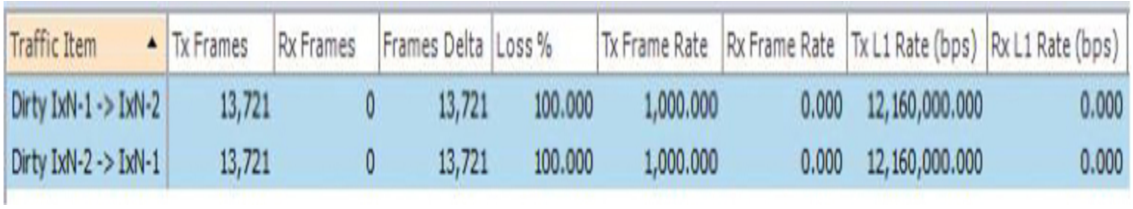
The first output shown in Figure 11-18 is from an IXIA traffic generator, where the initial view showcases the healthy state of the system, showing the same mirrored number of RX and TX frames on the connected ports.



Traffic Item ▲	Tx Frames	Rx Frames	Frames Delta	Loss %	Tx Frame Rate	Rx Frame Rate	Tx L1 Rate (bps)	Rx L1 Rate (bps)
Dirty IxN-1 -> IxN-2	154,659	154,659	0	0.000	1,000.000	1,000.000	12,160,000.000	12,192,000.000
Dirty IxN-2 -> IxN-1	154,659	154,659	0	0.000	1,000.000	1,000.000	12,160,000.000	12,128,000.000

Figure 11-18 *Output from IXIA in a Stable Topology*

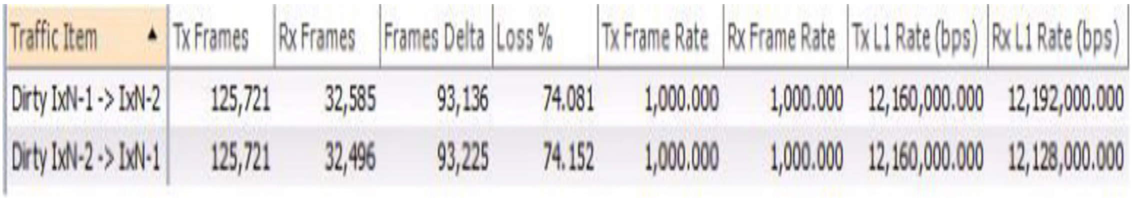
Upon the failure of the intermediate infrastructure, you can see the number of transmitted frames without any increasing RX packets, indicating a loss scenario in the network architecture that is under testing (see Figure 11-19).



Traffic Item ▲	Tx Frames	Rx Frames	Frames Delta	Loss %	Tx Frame Rate	Rx Frame Rate	Tx L1 Rate (bps)	Rx L1 Rate (bps)
Dirty IxN-1 -> IxN-2	13,721	0	13,721	100.000	1,000.000	0.000	12,160,000.000	0.000
Dirty IxN-2 -> IxN-1	13,721	0	13,721	100.000	1,000.000	0.000	12,160,000.000	0.000

Figure 11-19 *Active Scenario Where Packets Are Being Dropped in an IXIA*

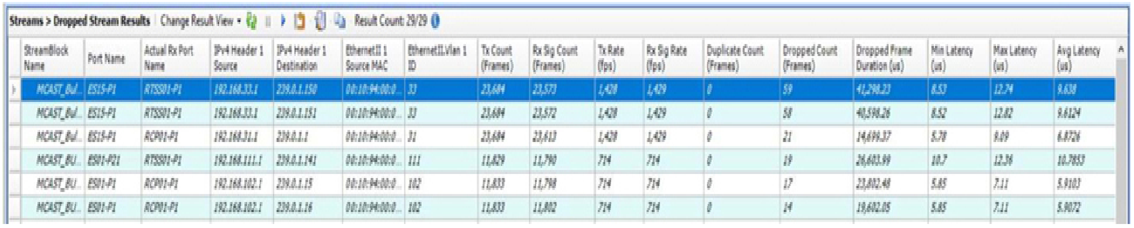
Figure 11-20 shows the network in recovery, where the RX frames are once again being received; however, there is an overall delta in lost packets being shown as a result of the outage.



Traffic Item ▲	Tx Frames	Rx Frames	Frames Delta	Loss %	Tx Frame Rate	Rx Frame Rate	Tx L1 Rate (bps)	Rx L1 Rate (bps)
Dirty IxN-1 -> IxN-2	125,721	32,585	93,136	74.081	1,000.000	1,000.000	12,160,000.000	12,192,000.000
Dirty IxN-2 -> IxN-1	125,721	32,496	93,225	74.152	1,000.000	1,000.000	12,160,000.000	12,128,000.000

Figure 11-20 *Recovering State on IXIA GUI Where the Network Is Converging*

Looking at the Spirent Traffic generator output in Figure 11-21, you can see a further metric, which is the loss duration in combination with the dropped frame count. This loss duration time is important because it provides a view on how rapidly the network is recovering and what the aggregate loss was, as perceived by the traffic generator. Having such data sets on hand allows a network team to properly tweak, tune, and optimize the architecture to achieve the lowest convergence times.

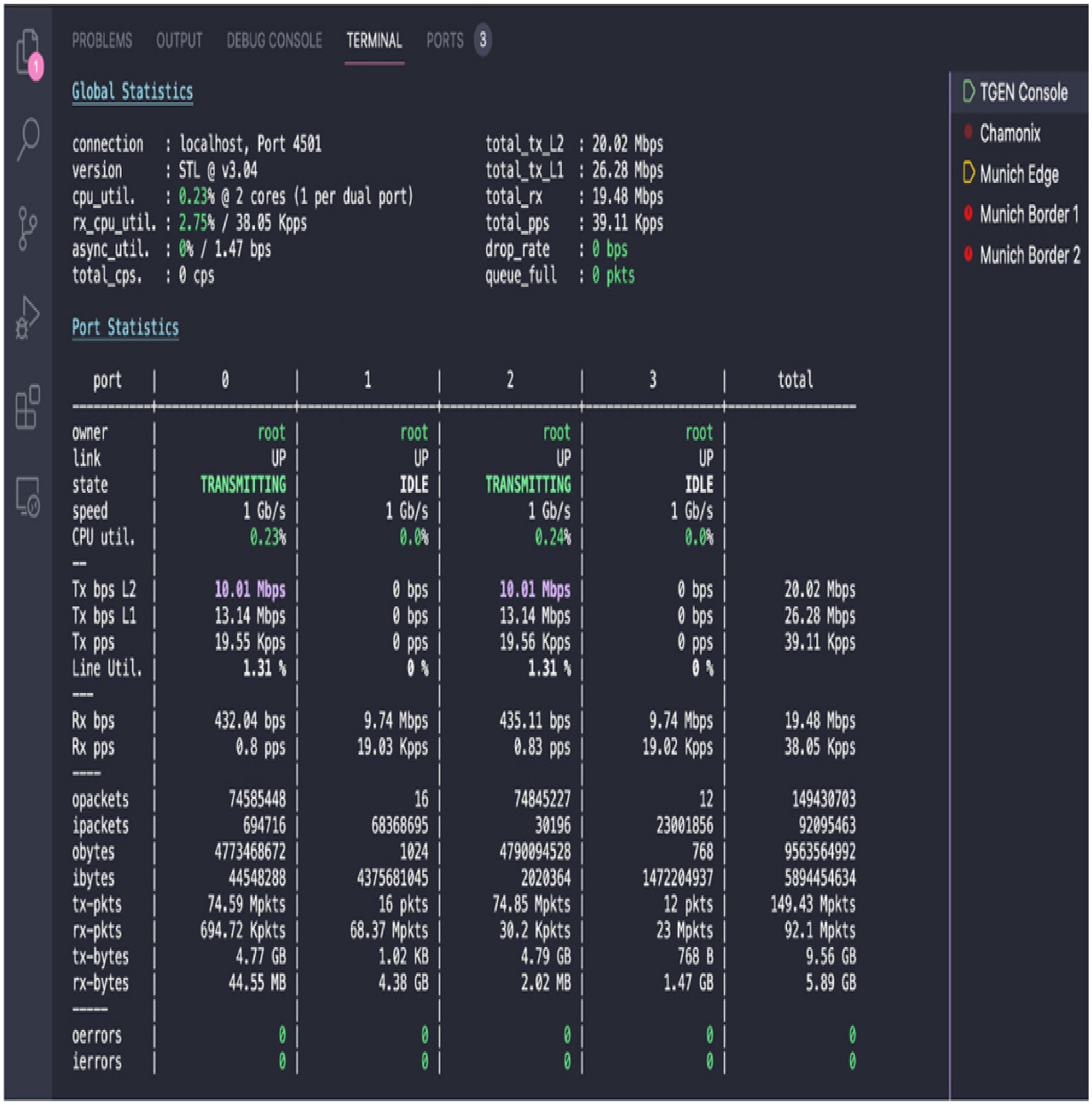


StreamBlock Name	Port Name	Actual Rx Port Name	IPv4 Header 1 Source	IPv4 Header 1 Destination	EthernetII 1 Source MAC	EthernetII.Vlan 1 ID	Tx Count (Frames)	Rx Sig Count (Frames)	Tx Rate (fps)	Rx Sig Rate (fps)	Duplicate Count (Frames)	Dropped Count (Frames)	Dropped Frame Duration (us)	Min Latency (us)	Max Latency (us)	Avg Latency (us)
MCAST_Bd...	ES15-P1	RTSS01-P1	192.168.33.1	239.0.1.150	00:10:94:00:0...	33	23,684	23,573	1,428	1,429	0	59	41,298.23	8.53	12.74	9.638
MCAST_Bd...	ES15-P1	RTSS01-P1	192.168.33.1	239.0.1.151	00:10:94:00:0...	33	23,684	23,572	1,428	1,429	0	58	40,598.26	8.52	12.82	9.6124
MCAST_Bd...	ES15-P1	RCP01-P1	192.168.31.1	239.0.1.1	00:10:94:00:0...	31	23,684	23,613	1,428	1,429	0	21	14,699.37	5.78	9.09	6.8726
MCAST_BU...	ES01-P21	RTSS01-P1	192.168.111.1	239.0.1.141	00:10:94:00:0...	111	11,829	11,790	714	714	0	19	26,603.99	10.7	12.36	10.7853
MCAST_BU...	ES01-P1	RCP01-P1	192.168.102.1	239.0.1.15	00:10:94:00:0...	102	11,833	11,798	714	714	0	17	23,802.48	5.85	7.11	5.9103
MCAST_BU...	ES01-P1	RCP01-P1	192.168.102.1	239.0.1.16	00:10:94:00:0...	102	11,833	11,802	714	714	0	14	19,602.05	5.85	7.11	5.9072

Figure 11-21 *Packet Loss Duration in Spirent*

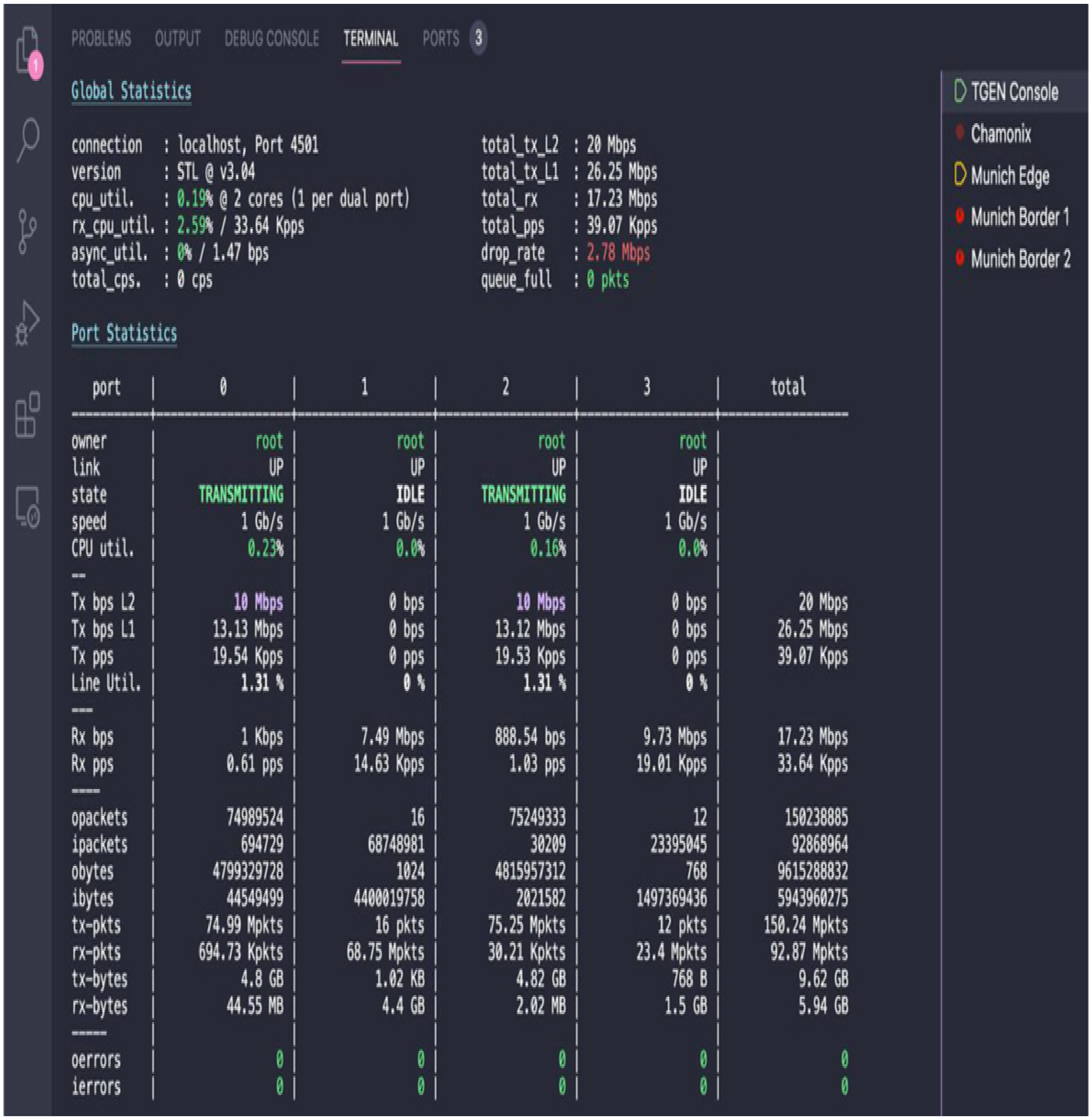
Depending on which traffic generation platform is chosen, you might need to procure specific, or under certain circumstances, special hardware line cards to transmit and receive in a stateful operation mode.

The views shown in Figure 11-22 and Figure 11-23 are from the third (and final) traffic generator that is being showcased, TRex. The advantage to the TRex traffic generator is that it is free and open source, which allows for a cheaper entry point for consumers who want to be able to perform more advanced validation activities. Given that it is an open-source project, you also have the ability to contribute to it and enrich it with enhancements that may be needed for your organization relatively easily.



```
Global Statistics

connection   : localhost, Port 4501                  total_tx_L2  : 20.02 Mbps
version      : STL @ v3.04                           total_tx_L1  : 26.28 Mbps
cpu_util.    : 0.23% @ 2 cores (1 per dual port)     total_rx     : 19.48 Mbps
rx_cpu_util. : 2.75% / 38.05 Kpps                    total_pps    : 39.11 Kpps
async_util.  : 0% / 1.47 bps                         drop_rate    : 0 bps
total_cps.   : 0 cps                                 queue_full   : 0 pkts

Port Statistics

   port   |       0       |       1       |       2       |       3       |     total
----------+---------------+---------------+---------------+---------------+----------------
owner     |          root |          root |          root |          root |
link      |            UP |            UP |            UP |            UP |
state     |  TRANSMITTING |          IDLE |  TRANSMITTING |          IDLE |
speed     |        1 Gb/s |        1 Gb/s |        1 Gb/s |        1 Gb/s |
CPU util. |         0.23% |          0.0% |         0.24% |          0.0% |
--        |               |               |               |               |
Tx bps L2 |    10.01 Mbps |         0 bps |    10.01 Mbps |         0 bps |     20.02 Mbps
Tx bps L1 |    13.14 Mbps |         0 bps |    13.14 Mbps |         0 bps |     26.28 Mbps
Tx pps    |    19.55 Kpps |         0 pps |    19.56 Kpps |         0 pps |     39.11 Kpps
Line Util.|        1.31 % |           0 % |        1.31 % |           0 % |
---       |               |               |               |               |
Rx bps    |    432.04 bps |     9.74 Mbps |    435.11 bps |     9.74 Mbps |     19.48 Mbps
Rx pps    |       0.8 pps |    19.03 Kpps |      0.83 pps |    19.02 Kpps |     38.05 Kpps
----      |               |               |               |               |
opackets  |      74585448 |            16 |      74845227 |            12 |      149430703
ipackets  |        694716 |      68368695 |         30196 |      23001856 |       92095463
obytes    |    4773468672 |          1024 |    4790094528 |           768 |     9563564992
ibytes    |      44548288 |    4375681045 |       2020364 |    1472204937 |     5894454634
tx-pkts   |   74.59 Mpkts |       16 pkts |   74.85 Mpkts |       12 pkts |   149.43 Mpkts
rx-pkts   |  694.72 Kpkts |   68.37 Mpkts |    30.2 Kpkts |      23 Mpkts |     92.1 Mpkts
tx-bytes  |       4.77 GB |       1.02 KB |       4.79 GB |         768 B |        9.56 GB
rx-bytes  |      44.55 MB |       4.38 GB |       2.02 MB |       1.47 GB |        5.89 GB
-----     |               |               |               |               |
oerrors   |             0 |             0 |             0 |             0 |              0
ierrors   |             0 |             0 |             0 |             0 |              0
```

Figure 11-22 *TRex Transmitting Data Without Dropping Any Data*



```
Global Statistics

connection   : localhost, Port 4501                  total_tx_L2  : 20 Mbps
version      : STL @ v3.04                           total_tx_L1  : 26.25 Mbps
cpu_util.    : 0.19% @ 2 cores (1 per dual port)     total_rx     : 17.23 Mbps
rx_cpu_util. : 2.59% / 33.64 Kpps                    total_pps    : 39.07 Kpps
async_util.  : 0% / 1.47 bps                         drop_rate    : 2.78 Mbps
total_cps.   : 0 cps                                 queue_full   : 0 pkts

Port Statistics

   port    |       0         |       1         |       2         |       3         |      total
-----------+-----------------+-----------------+-----------------+-----------------+-----------------
owner      |            root |            root |            root |            root |
link       |              UP |              UP |              UP |              UP |
state      |    TRANSMITTING |            IDLE |    TRANSMITTING |            IDLE |
speed      |          1 Gb/s |          1 Gb/s |          1 Gb/s |          1 Gb/s |
CPU util.  |           0.23% |            0.0% |           0.16% |            0.0% |
--         |                 |                 |                 |                 |
Tx bps L2  |         10 Mbps |           0 bps |         10 Mbps |           0 bps |         20 Mbps
Tx bps L1  |      13.13 Mbps |           0 bps |      13.12 Mbps |           0 bps |      26.25 Mbps
Tx pps     |      19.54 Kpps |           0 pps |      19.53 Kpps |           0 pps |      39.07 Kpps
Line Util. |          1.31 % |             0 % |          1.31 % |             0 % |
---        |                 |                 |                 |                 |
Rx bps     |          1 Kbps |       7.49 Mbps |      888.54 bps |       9.73 Mbps |      17.23 Mbps
Rx pps     |        0.61 pps |      14.63 Kpps |        1.03 pps |      19.01 Kpps |      33.64 Kpps
----       |                 |                 |                 |                 |
opackets   |        74989524 |              16 |        75249333 |              12 |       150238885
ipackets   |          694729 |        68748981 |           30209 |        23395045 |        92868964
obytes     |      4799329728 |            1024 |      4815957312 |             768 |      9615288832
ibytes     |        44549499 |      4400019758 |         2021582 |      1497369436 |      5943960275
tx-pkts    |     74.99 Mpkts |         16 pkts |     75.25 Mpkts |         12 pkts |    150.24 Mpkts
rx-pkts    |    694.73 Kpkts |     68.75 Mpkts |     30.21 Kpkts |      23.4 Mpkts |     92.87 Mpkts
tx-bytes   |          4.8 GB |         1.02 KB |         4.82 GB |           768 B |         9.62 GB
rx-bytes   |        44.55 MB |          4.4 GB |         2.02 MB |          1.5 GB |         5.94 GB
-----      |                 |                 |                 |                 |
oerrors    |               0 |               0 |               0 |               0 |               0
ierrors    |               0 |               0 |               0 |               0 |               0
```

Figure 11-23 *Packet Drops Observed in TREX*

Monitoring Security Convergence

Within network domains that use macro- and microsegmentation, the convergence of security measures is handled via two separate methods. Macrosegmentation, which is achieved via routing table separation by placing systems and endpoints in different VRFs, is achieved via the routing protocols themselves. Using protocols like BFD can help speed up such convergence activities. For microsegmentation, however, you need to look at an array of other protocols when assessing the convergence of security group tag information and corresponding ACLs.

To understand convergence of security group tag information, let’s revisit how the data is presented and seen on Cisco routers and switches. SGT information is maintained in a binding table, which maps the corresponding SGT to its respective IP prefix. The sourcing protocol of the binding entry can be seen in the source field of Example 11-11.

Example 11-11 *IP Address to SGT and Corresponding Group Name and Source Verification*

```
### Viewing the details on a specific binding:

Edge#show cts role-based sgt-map vrf ENT  172.16.60.120  details
Active IPv4-SGT Bindings Information

IP Address          Security Group                              Source
==================================================================
172.16.60.120       1120:muc_tgen_120                            SXP
Edge#

### Verifying how many SGT bindings exist on a switch for the VRF

Edge#show cts role-based sgt-map vrf ENT all | count [0-9]\.|[A-F
Number of lines that match regexp = 161
Edge#
```

Alternatively, you can query this through an API call when using RESTCONF on the switches, as shown in Example 11-12.

Example 11-12 *Basic RESTCONF Python Script Querying TrustSec Parameters*

```
import requests
```

```
url = "https://<Switch IP Address>:443/restconf/data/Cisco-IOS-XE

payload = {}
headers = {
  'Connection': 'keepalive',
  'Accept': 'application/yang-data+json',
  'Authorization': 'Basic <removed>'
}

response = requests.request("GET", url, headers=headers, data=pay

print(response.text)
```

When you’re looking at convergence, the priority of the source is an important consideration to remember. Much like administrative distance in routing domains, the priority of the source of the SGT information will play a role in whether a backup mapping is potentially used. Much like hold time is a factor in IGP domains, for protocols like SXP that are responsible for exchanging IP-to-SGT bindings, similar timers can be configured to enhance the time that entries are preserved.

Priority order for Cisco TrustSec:

1. VLAN
2. CLI
3. L3IF
4. LISP_REMOTE_HOST
5. LISP_LOCAL_HOST
6. OMP
7. SXP
8. ARP
9. LOCAL

10. CACHING
11. INTERNAL

In addition to the binding being present to map the IP address to a security group tag, you need to retrieve and download corresponding SGACLs to ensure that the appropriate enforcement can take place.

This overview can be retrieved as shown in Example 11-13 through Example 11-15.

Example 11-13 *View of Downloaded SGTs on a Switch*

```
Edge#show cts rbacl
CTS RBACL Policy
================
RBACL IP Version Supported: IPv4 & IPv6
  name   = Deny_IP_Log-00
  IP protocol version = IPV4, IPV6
  refcnt = 2
  flag   = 0xC1000000
  stale  = FALSE
  RBACL ACEs:
    deny ip log

  name   = PermitDHCP-03
  IP protocol version = IPV4
  refcnt = 5
  flag   = 0x41000000
  stale  = FALSE
  RBACL ACEs:
    permit udp dst eq 67
    permit udp src eq 68
    permit udp dst eq 68
    permit udp src eq 67
    permit udp src eq 53
    permit udp dst eq 53
    deny ip log-input
```

Example 11-14 *Count of Total Downloaded SGACLs*

```
Edge#show cts rbacl | count name
Number of lines that match regexp = 46
Edge#
```

Example 11-15 *Mappings Between Source and Destination Groups and Their Corresponding ACLs*

```
Edge#show cts role-based permissions from 1120
IPv4 Role-based permissions from group 1120:muc_tgen_120 to group
        SRC1120DST120-00
RBACL Monitor All for Dynamic Policies : FALSE
RBACL Monitor All for Configured Policies : TRUE
Edge#
```

In addition to SGTs binding information, arriving through a number of different sources that have different levels of priority, when you're using SXP, in some scenarios the binding for the same prefix may exist multiple times, as learned via different peerings. Verifying convergence and installation of redundant entries upon the failure of the upstream infrastructure is one test that you can perform to confirm that the northbound peerings are functional.

When you're configuring SXP as a means for IP-to-SGT binding reflection, the recommendation is to use a north-to-south approach. What this essentially means is that the propagation of SGT to prefix mappings should take place vertically and diagonally rather than horizontally, as depicted in Figure 11-24.

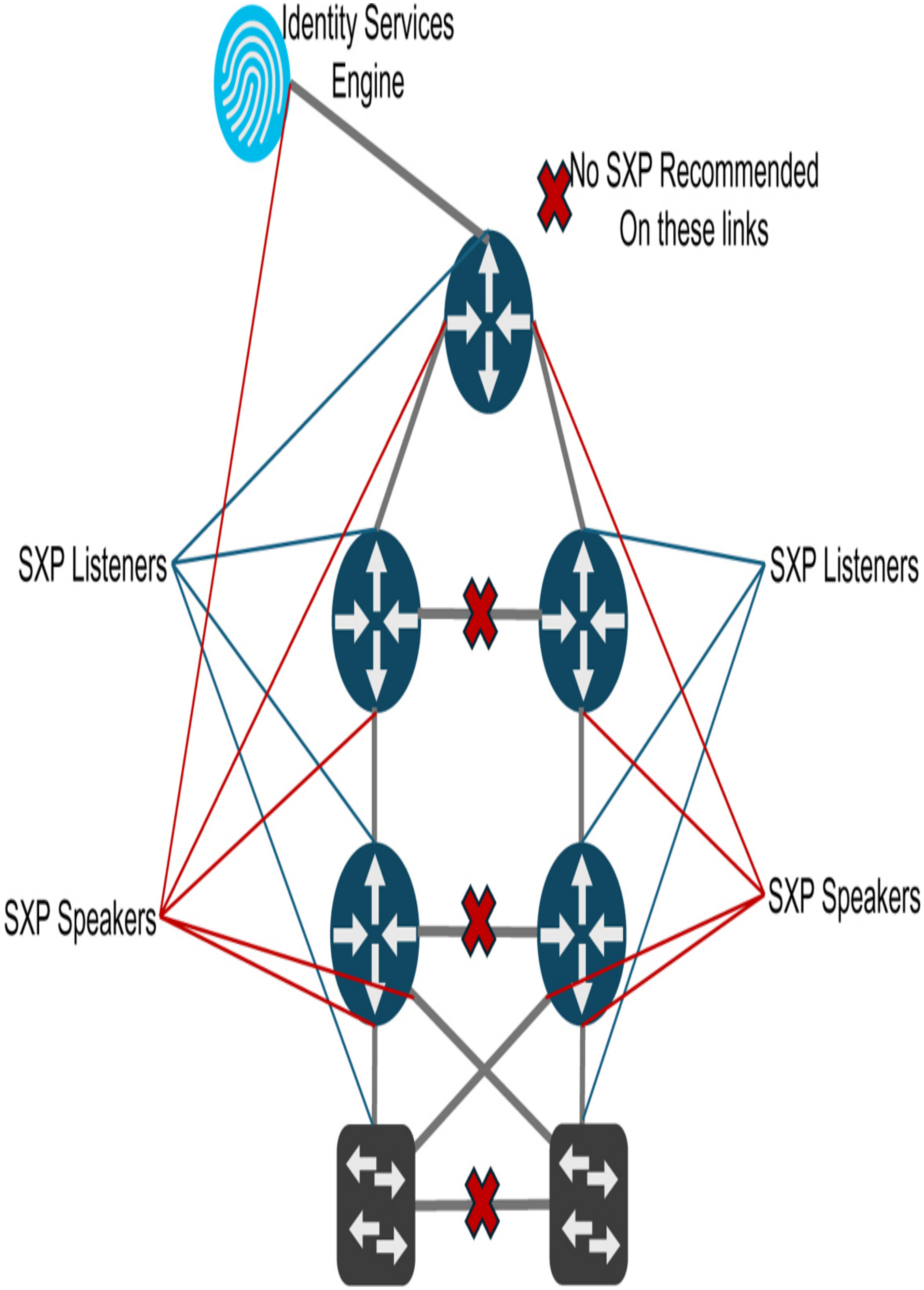


Figure 11-24 *Hierarchical SXP Peering Structure*

Example 11-16 provides a view of the peerings that are performed between nodes and an example of redundant binding mappings that are being learned from upstream peers.

Example 11-16 *Redundant SXP Learned Bindings*

```
Edge#show cts sxp connections brief
 SXP              : Enabled
 Highest Version Supported: 5
 Default Password : Set
 Default Key-Chain: Not Set
 Default Key-Chain Name: Not Applicable
 Default Source IP: Not Set
Connection retry open period: 120 secs
Reconcile period: 120 secs
Retry open timer is not running
Peer-Sequence traverse limit for export: Not Set
Peer-Sequence traverse limit for import: Not Set

-----------------------------------------------------------------
Peer_IP          Source_IP        Conn Status
-----------------------------------------------------------------
10.100.0.2       172.16.10.70     On
(dd:hr:mm:sec)
172.16.10.65     172.16.10.70     On
(dd:hr:mm:sec)

Total num of SXP Connections = 2

Edge# show cts sxp sgt-map vrf ENT

/172.16.60.120
filtering...
IPv4,SGT: <172.16.60.120 , 1120:muc_tgen_120>
```

```
source  : SXP;
Peer IP : 10.100.0.2;
Ins Num : 12;
Seq Num : 4787
Peer Seq: AC100C01,C0A8C1C1,C0A84665,
IPv4,SGT: <172.16.60.120 , 1120:muc_tgen_120>
source  : SXP;
Peer IP : 172.16.10.65;
Ins Num : 1;
Status  : Active; << The active status denotes that SXP binding e
Seq Num : 5061
Peer Seq: AC100C01,C0A8C1C1,C0A84665,
```

During a network convergence event, depending on the configured timers, the entries learned via SXP will be purged from the SGT table or will continue to be held. If networks encounter increased instability, or if there are known planned maintenance windows coming up and the devices rely on SXP for binding table updates, it is important that appropriate timers are in place to weather potential turbulence.

If there is a scenario where the hard outage of a node is responsible for sharing key binding information, the results of this outage could manifest themselves in the form of dropped packets because the local policy on the respective system does not have all the needed information about the respective mappings.

These drops, as shown in Example 11-17, can be seen through monitoring the HW-Denied entries on the TrustSec enforcement point.

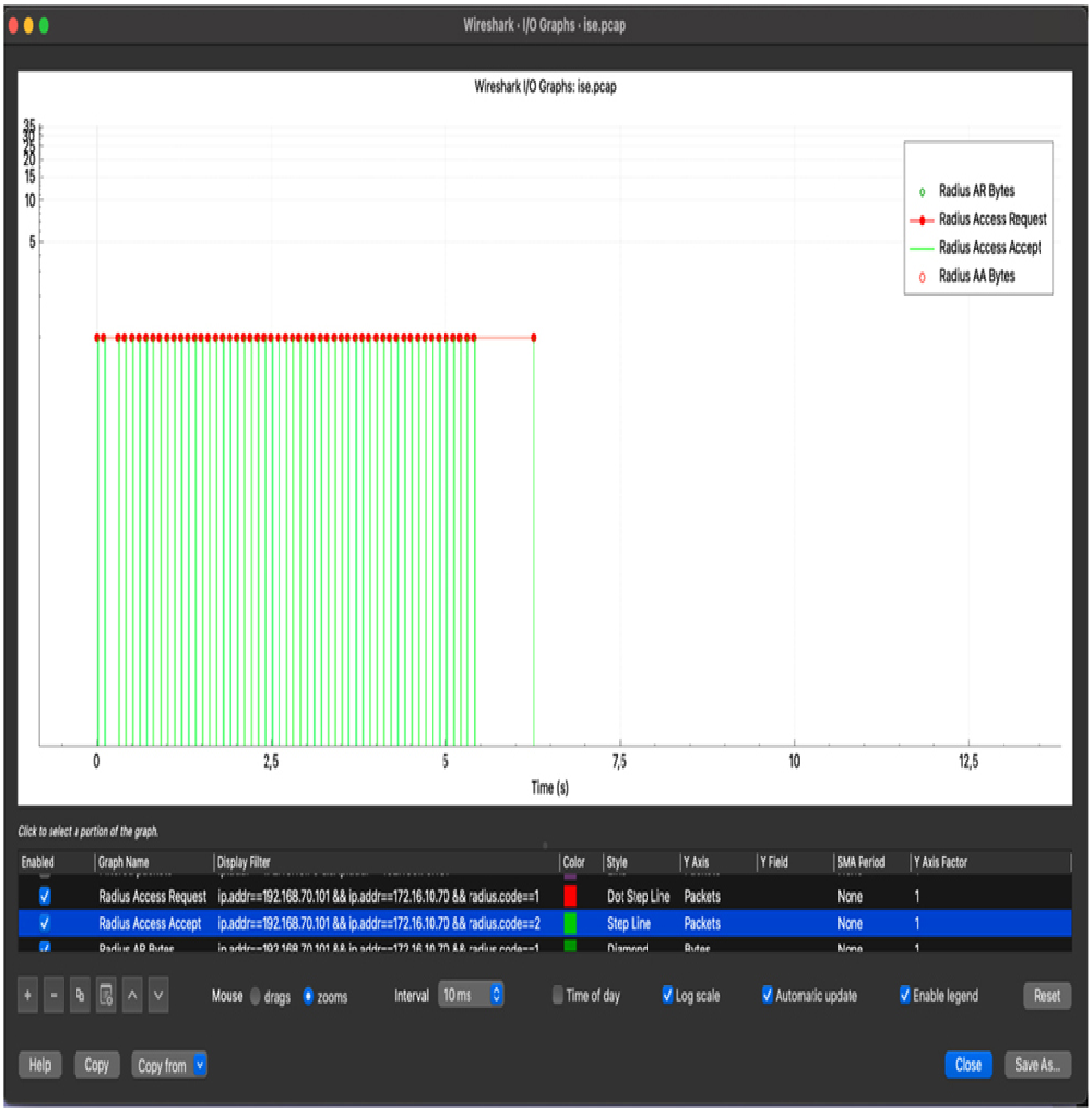
Example 11-17 *Permit and Deny SGT Enforcement Counters*

```
Branch-FIAB#show cts role-based counters
Role-based IPv4 counters
From     To       SW-Denied  HW-Denied  SW-Permitt HW-Permitt SW-Mo
*        *        0          323578     0          0          0
2        2        0          0          0          0          0
```

```
4       23      0      0          0      135791     0
2101    1101    0      0          0      18664508   0
2102    1102    0      27996762   0      0          0
2103    1103    0      0          0      9332254    0
2104    1104    0      0          0      18664508   0
2105    1105    0      531732     0      0          0
2106    1106    0      9332254    0      0          0
2107    1107    0      0          0      18664508   0
2108    1108    0      0          0      27996762   0
2109    1109    0      0          0      9332254    0
2110    1110    0      354488     0      0          0
2111    1111    0      0          0      27996762   0
```

In addition to SXP learned binding convergence, there are scenarios that can come into play when a system is reloading, resulting in the need for Radius to once again propagate the Cisco TrustSec Environmental Data (SGTs) and the respective security policy information (RBACLS).

Figure 11-25 shows the time taken to download retrieve CTS policy rulesets from the Radius architecture.



Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis Factor
☑	Radius Access Request	ip.addr==192.168.70.101 && ip.addr==172.16.10.70 && radius.code==1	🟥	Dot Step Line	Packets		None	1
☑	Radius Access Accept	ip.addr==192.168.70.101 && ip.addr==172.16.10.70 && radius.code==2	🟩	Step Line	Packets		None	1
☑	Radius AR Bytes	ip.addr==192.168.70.101 && ip.addr==172.16.10.70 && radius.code==1	🟩	Diamond	Bytes		None	1

Figure 11-25 *Policy Retrieval Time Shown in Wireshark I/O Graphs*

The time it takes to retrieve this information comes down to the number of bindings that require propagation to the respective switch or router from ISE, in addition to the retrieval of all relevant SGACLs that may be required. The more disparate workloads and systems that connect to a switch or switch stack, the larger the list of role-based ACLs that will require retrieval.

Summary

In modern networks, convergence continues to remain a key and critical aspect of running a stable environment. While there have been significant improvements in protocols, timers, and mechanisms to converge, considering additional factors, including segmentation policy and security enforcement, is important in this problem space. If planning and deployment structure are approached correctly, rapid convergence can be achieved for both network and security concepts.

Part 3: Deployment Best Practices, Architectures, and Use Cases

Chapter 12. Software-Defined Network Deployment Best Practices

In this chapter, you will learn about the following:

- SDN deployment challenges
- SDN deployment planning and design best practices
- SDN deployment migration best practices
- SDN deployment operations best practices

Introduction

In today’s software-driven world, being able to deploy and maintain a network infrastructure without some form of software or tools managing it is unheard of. As networks become more complex due to the demands and needs of upper layer applications, some form of network orchestration is needed to provide better control, predictability, and consistency. The days of configuring networks by logging in to the device via SSH are nearing an end and are being replaced by solution-based network controllers or in-house built automation scripts run by Ansible or Terraform.

Software-defined networking (SDN) is a networking architecture that separates control functions from forwarding functions, allowing for greater automation and programmability. It enables enterprises to use a centralized network management server, called an SDN Controller (SDNC), to control network devices and access to applications.

In today's fast-paced business world, companies need their networks to be as agile as they are. SDN allows businesses to change size quickly, add new apps easily, and react to security issues immediately. This agility is crucial for keeping up with the rapid pace of change. Additionally, SDN provides enhanced visibility and control over network traffic, enabling businesses to see and manage their network more effectively.

While many SDN solutions are turnkey and their standalone implementation is relatively straightforward, deploying a completely new network is often not feasible for businesses due to an existing customer base and market conditions. As SDN adoption increases among network operators, there is a growing need for best practices in deployment.

In this chapter, we will explore various on-premises and cloud-hosted controllers, including Cisco Catalyst Center, APIC, Catalyst Manager, and Cisco Meraki Dashboard. These solutions are well-suited for different types of campus networks, data centers, and managed service providers. The journey toward full adoption of SDN can be challenging. Some of the key challenges include

- **Inadequate Knowledge of the Organization's Architecture:** Medium to large enterprises often face significant blind spots in their IT infrastructure. IT departments typically operate in functional modules—such as Applications, Network, Security, and Cloud—because managing all aspects of IT can be overwhelming for any single person or team. While this modular structure enhances agility and ensures smooth operations, it can also create silos that result in visibility gaps between departments and processes. SDN deployments effectively break down these silos by integrating various functions into centralized controllers, significantly improving visibility across Applications, Network, and Security.

- **Legacy Network Constraints:** Refreshing network hardware and updating software that are prerequisites for SDN deployment are common tasks and are generally considered straightforward. However, migrating an existing network may require adapting SDN to the inherent constraints of the legacy design. These constraints can include IP addressing schemes, VLAN configurations, and existing security policies. Therefore, careful consideration and planning of these factors are essential for a successful integration.

- **Lack of Automation:** One of the primary advantages of SDN is the opportunity for increased network automation. SDN provides workflows that automate the network deployment and operations at scale. Failing to utilize automation tools can lead to manual configuration overhead, operational complexity, and slower service provisioning. Embracing automation is essential for realizing SDN's full potential.
- **Troubleshooting Complexity:** The integration of SDN overlays and the interaction with legacy control planes can enhance the network’s capabilities, though it also introduces some complexities in troubleshooting. While SDN workflows streamline deployment, they utilize advanced protocols like LISP, VXLAN, and OMP, which may be less familiar to network administrators who are experienced with traditional IGP and BGP protocols. Embracing this shift presents an opportunity for growth and skill development. By investing in comprehensive training and education, organizations can equip their teams with the knowledge needed to navigate these challenges, ultimately ensuring a successful implementation of SDN.
- **Security Oversights:** The SDN controller requires a high level of trust because it serves as the central brain for the entire network. Its placement, whether in the cloud or on-premises, introduces new security considerations that must be addressed during the design phase. Neglecting to account for controller vulnerabilities, using insecure communication protocols, and implementing inadequate access control mechanisms can expose the network to security breaches and compromise sensitive data.

Network Deployment Lifecycle

Before we explore the best practices for deploying software-defined networking, let's familiarize ourselves with the current network lifecycle and how SDN can align with it. The PPDIOO model, proposed by Cisco, outlines a continuous lifecycle for network services. PPDIOO stands for Prepare, Plan, Design, Implement, Operate, and Optimize. Each phase represents an effort that focuses on a set of use cases to ensure the network deployment remains consistent, regardless of the SDN controller or technology being used. The PPDIOO phases are as follows:

- **Prepare:** This phase involves establishing the organizational requirements, developing a network strategy, and proposing a high-level conceptual architecture identifying technologies that can best support the architecture. The prepare phase can establish a financial justification for network strategy by assessing the business case for the proposed architecture.

- **Plan:** The plan phase involves identifying initial network requirements based on goals, facilities, user needs, and so on. This phase involves characterizing sites and assessing any existing networks and performing a gap analysis to determine whether the existing system infrastructure, sites, and the operational environment can support the proposed system. A project plan is useful for helping manage the tasks, responsibilities, critical milestones, and resources required to implement changes to the network. The project plan should align with the scope, cost, and resource parameters established in the original business requirements.

- **Design:** The initial requirements that were derived in the planning phase drive the activities of the network design specialists. The network design specification is a comprehensive detailed design that meets current business and technical requirements, and incorporates specifications to support availability, reliability, security, scalability, and performance. The design specification is the basis for the implementation activities.

- **Implement:** In this phase, the network is built or additional components are incorporated according to the design specifications, with the goal of integrating devices without disrupting the existing network or creating points of vulnerability.

- **Operate:** Operation is the final test of the appropriateness of the design. The operational phase involves maintaining network health through day-to-day operations, including maintaining high availability and reducing expenses. The fault detection, correction, and performance monitoring that occur in daily operations provide the initial data for the optimization phase.

- **Optimize:** The optimize phase involves proactive management of the network. The goal of proactive management is to identify and resolve issues before they affect the organization. Reactive fault detection and correction (troubleshooting) are needed when proactive management cannot predict and mitigate failures. In the PPDIOO process, the optimization phase can prompt a network redesign if too many network problems and errors arise, if performance does not meet expectations, or if new applications are identified to support organizational and technical requirements.

The PPDIOO framework is commonly used in large enterprises, where networking teams are organized into Design, Implementation, and Operations to cover the entire PPDIOO lifecycle. However, this process is agnostic to technology or the size of the network that is being deployed; hence, it can help any organization embarking on a transformation journey of its own. Aligning SDN deployment with this existing structure can help avoid friction and ensure a smoother implementation. Let’s evaluate best practices at each stage of this process.

Stage 1: Planning and Design

The prepare and planning phases are crucial for successful SDN deployment because it directly connects business outcomes to the technical design of the network. By establishing a clear link between these outcomes and the network's technical specifications, organizations can ensure a successful implementation that delivers tangible benefits and aligns with their overall strategic goals. This phase serves as a blueprint for this alignment, laying the foundation for a robust, secure, and agile SDN environment. In the following sections, we will review some key deliverables from this phase and discuss best practice deployment recommendations

Defining Network Requirements and Use Cases

A fundamental step in the planning process involves a comprehensive assessment of the organization’s specific needs and objectives. When an organization is embarking on an SDN adoption journey, it's paramount to clearly define your network pain points and identify the specific use cases where SDN can make a tangible difference. This initial planning phase lays the foundation for a successful and impactful implementation. Some key considerations are

- **Business Drivers and Technical Objectives:** The decision to adopt software-defined networking should be guided by clear business needs and technical objectives. It's essential to align SDN's capabilities with your organization's strategic goals. Business needs may arise from customers, internal stakeholders, or simply from growth and expansion plans. Occasionally, urgent requirements may emerge due to security breaches, which can render the current network ineffective. It's important to classify these business requirements to determine if they align with long-term strategies or are simply short-term responses to immediate problems. Once clearly documented, these needs should be broken down into specific technical requirements that will drive the SDN deployment options.

- **Improving Network Agility:** Software-defined networking introduces a new level of agility to networks, enabling organizations to quickly adapt to changing requirements and network conditions. By centralizing control and utilizing software-based management, SDN allows for dynamic adjustments to network configurations and policies. This flexibility is especially beneficial for organizations with distributed networks because it facilitates efficient traffic routing, optimization, and resource allocation across various locations. SDN’s ability to simplify underlying network complexities and automate operations empowers organizations to proactively respond to evolving demands, optimize application performance, and enhance security throughout the network.

- **Network Segmentation:** Organizations have effectively used network segmentation for many years through VLANs, VRFs, and zone-based firewalls. As networks expand, it’s important to address challenges related to scalability and management, such as handling numerous VRFs, reaching the limit of 4094 unique VLANs, and managing complex ACLs. By adopting strategies that streamline identity management beyond just relying on IP addresses, organizations can create a more efficient and manageable network environment. SDN excels in providing granular segmentation capabilities. It allows organizations to create isolated network segments for different user groups, departments, or applications. This approach is particularly beneficial for enhancing security and ensuring compliance in industries with strict regulatory requirements.

- **Improved Security and Observability:** SDN's centralized control and programmability facilitate the implementation of fine-grained security policies. Microsegmentation, a key SDN security capability, enables the isolation of network traffic at a granular level, limiting the impact of security breaches and enhancing overall network security. The SDN solutions present an exciting advancement in observability by moving beyond traditional methods like SNMP and syslog. By integrating streaming telemetry through NETCONF and gRPC, these solutions make robust observability easily accessible right out of the box. Additionally, the centralized controllers enhance data analysis by providing the computational resources needed to contextualize information, ultimately delivering deeper insights into system performance.

By defining well-articulated use cases, you can ensure that your SDN adoption is focused, measurable, and aligned with your organization’s overall goals.

Selecting the Right SDN Architecture

When you are selecting an SDN architecture, it is important to consider the various deployment models and their impact on factors such as scalability, performance, and resilience. Cisco provides multiple options that can be categorized into two primary groups.

On-Premises Controller

In the on-premises controller model, the responsibility falls on the customer to manage the infrastructure and build a resilient cluster for SDN controllers. This effort involves configuring hardware, ensuring optimal performance, and maintaining system integrity. The customer is also tasked with crucial operational duties, including upgrading software to keep the system current, performing regular backups to safeguard data, and implementing a disaster recovery plan to quickly restore service in the event of an outage. A reliable Internet connection to the vendor’s cloud infrastructure is essential because it enables the customer to easily access updates and patches. This connection not only streamlines maintenance but also enhances the system's overall security by allowing for timely vulnerability fixes. Moreover, one of the significant advantages of connecting to the cloud is the ability to leverage various artificial intelligence and machine learning models that the provider offers. Cisco Catalyst Center exemplifies this capability, featuring tools for endpoint profiling that enhance network visibility and security. It also incorporates AI-driven radio resource management (RRM) to optimize wireless performance dynamically, ensuring a seamless user experience. Additionally, strong network security measures are reinforced by TALOS, a threat intelligence service that provides real-time protection against known vulnerabilities and emerging threats. By utilizing these advanced features, customers can achieve a more robust, efficient, and secure networking environment. Figure 12-1 shows an example of an on-premises controller for Cisco’s SD-Access solution.

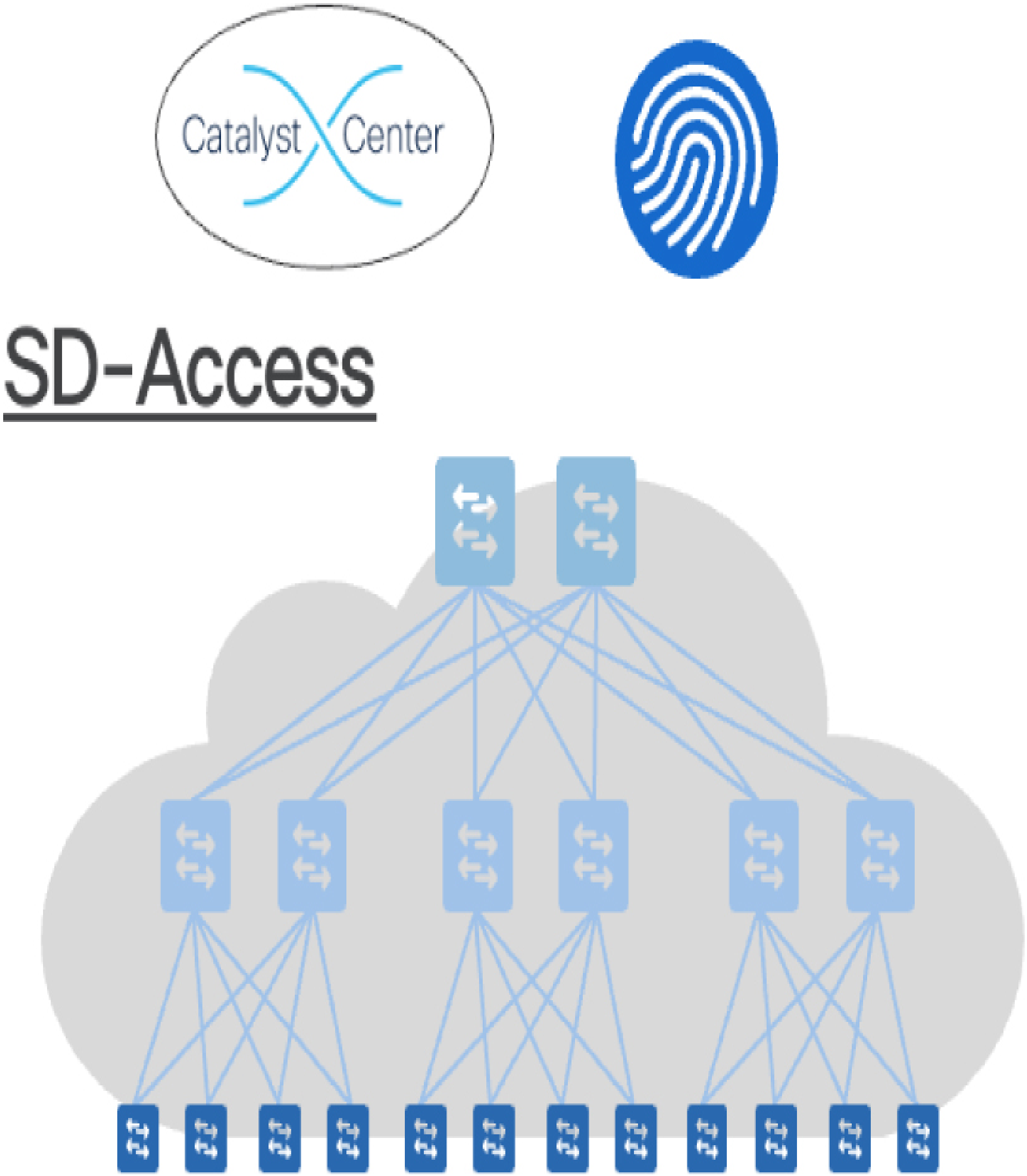


Figure 12-1 *On-premises Controller: Cisco Catalyst Center*

Software-as-a-Service (SaaS) Controllers

In the SaaS controller model, the vendor assumes responsibility for managing the entire infrastructure provisioning process and ensuring seamless operations. This effort includes committing to a service-level agreement (SLA) that guarantees a specified level of availability for the services provided. For example, Cisco has two offerings that are relevant in this context: the SD-WAN Catalyst Manager and the Meraki Dashboard. These tools are designed to facilitate efficient networking and management within the cloud framework. However, it is important to note that customers do bear some responsibilities in this arrangement. Specifically, they are tasked with configuring their networks by opening the necessary ports to allow communication with the cloud controller from their on-premises infrastructure. Additionally, the implementation of AI and ML workflows can be available based on the customer's choice of licensing. Typically, access to these advanced features may require purchasing a higher-tier subscription, which provides enhanced capabilities and support for more sophisticated operational needs. Figure 12-2 shows a sample SaaS-based controller for Cisco’s SD-WAN solution.

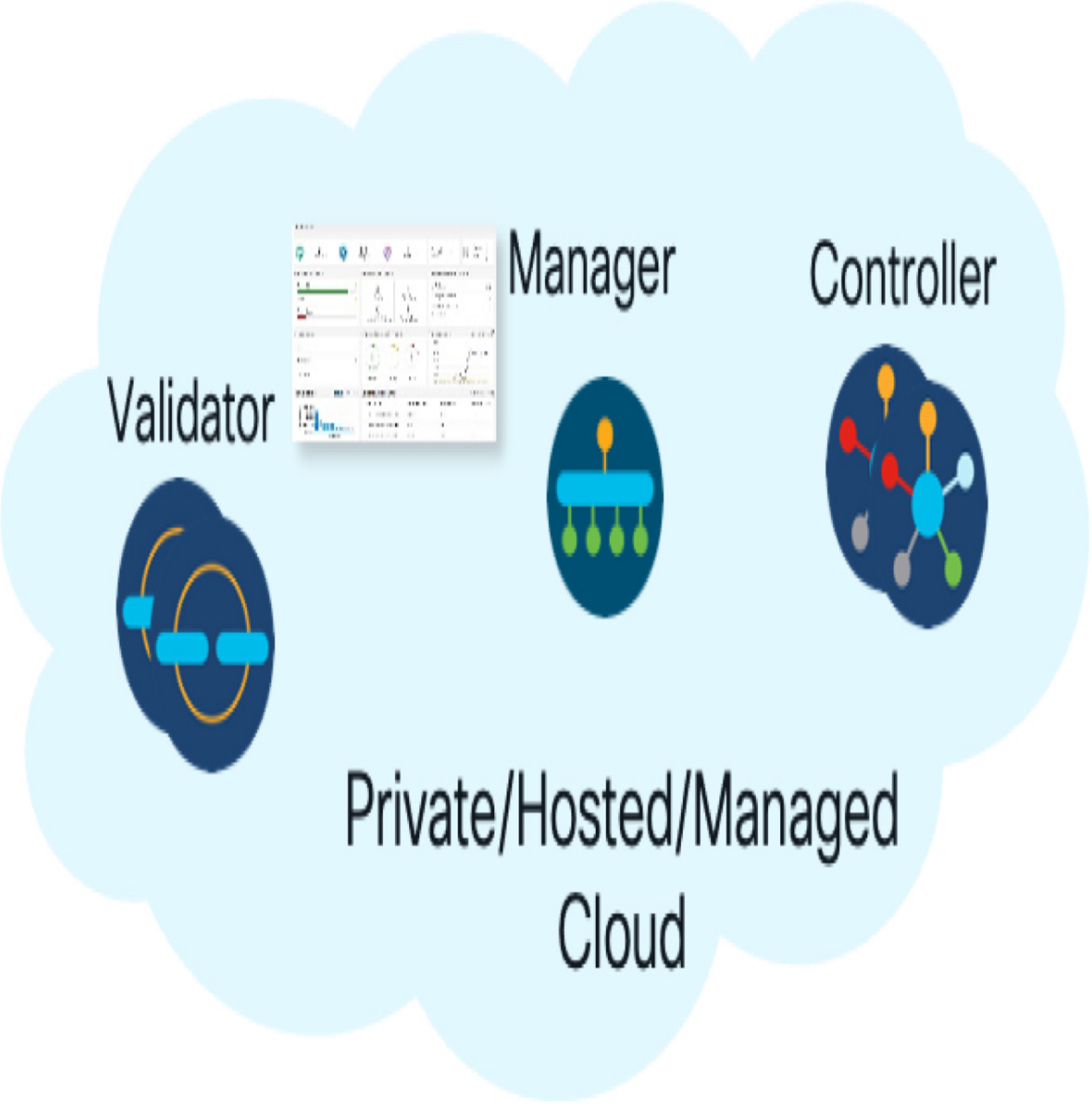


Figure 12-2 *SaaS Controller: Cisco Catalyst Manager*

Reviewing the Key Characteristics of SDN Controllers

Regardless of the controller type, on-premises or the cloud, SDN controllers share several common characteristics that are essential or deemed important:

- A clean, intuitive, and intelligent graphical user interface, providing all information in a single view.
- Controllers that do not handle the data path of actual traffic. This means that in the event of a controller outage, data flow remains unaffected; however, new policy provisioning will be unavailable during the outage.
- The ability to scale easily with significant design changes and support for business continuity through Active/Active or Automated/Transparent Active/Standby disaster recovery.
- An API-first approach to enable scalability and reusable workflows for automation.
- Desired integration with third-party operation and management tools.
- Compliance with data storage, retention, and privacy policies based on the regulations of each country.
- Clearly documented licensing requirements for features, readily available to customers for effective feature planning.
- Clear outlines of maximum scalability, any latency requirements for connecting to controllers, ports that need to be open to connect to the vendor's cloud, and specifics on any provided AI/ML models.

Designing a Robust Network Topology

The first step in deployment is planning the physical and logical placement of SDN controllers, switches, routers, and other network components.

While some design recommendations may be available from vendors, effective planning requires a solid understanding of the enterprise environment. The architect must decide on the physical placement based on scale and compliance, such as whether to use a single cluster or a distributed cluster, and whether to implement the solution in the cloud or on-premises. This is followed by the creation of a logical design document that outlines the ecosystem of network devices and controllers. For organizations with existing networks, this process presents an opportunity to optimize and consolidate their designs rather than simply adding controllers to their current setup. Here are some of Cisco's recommendations for the placement of SDN controllers.

- **Cisco's Catalyst Manager:** This tool is used in a multiregion SD-WAN deployment, to ensure resilience, whether they are on-premises or in the cloud. For high availability, a three-node Catalyst Manager cluster is recommended, with additional nodes added for scalability as needed. The Catalyst Manager controllers establish TLS/DTLS tunnels for provisioning, and controllers are installed in each region monitoring purposes.
- **Cisco's Catalyst Center:** This is a widely used tool for automated campus provisioning and monitoring. The Catalyst Center controllers are deployed in a cluster to achieve high availability, with a three-node cluster recommended for most deployments. These three-node clusters distribute services and enable automated service restoration, allowing services to be shifted from one node to another in the event of a failure. Furthermore, a disaster recovery architecture is implemented to provide automatic failover to a secondary data center cluster.

Once controller placement is established, focus on network redundancy and high availability. Use a hierarchical design in the underlay network for redundancy. The distribution layer should provide dual equal-cost paths to the core and access layer for quick recovery from link or node failures. Stacking access layer switches increases port density and ensures control plane redundancy.

The next step is to start planning the routing protocol. Almost all SDN technologies introduce the concept of overlay and underlay routing. The underlay network employs a Layer 3 routed access design, eliminating Layer 2 from distribution switches. This approach can improve performance and reduce complexity. The underlay connectivity is hidden from the rest of the user networks, and the overlay subnet primarily serves the users and devices in the network.

For overlay SDN solutions, some kind of routing protocol runs in the control plane. For example, Catalyst Manager runs OMP, and Catalyst Center runs LISP in the control plane or a well-known routing protocol like BGP. These protocols rely on the underlay protocol for reachability and use the overlay protocol to build user reachability information.

Bidirectional forwarding detection (BFD) provides subsecond failure detection for faster convergence and can run on top of BGP connections or Catalyst Manager's OMP routing. Please remember that most of the time control plane configuration and L3 routing are prescribed and automated by workflow; however, it is important to understand the implementation details due to interworking and connectivity outside of SDN Fabric. CHUCK

Another design consideration is maximum transmission units (MTUs). With overlay protocols, increased MTU sizes are common and can lead to potential mismatches, particularly when integrating SDN with legacy networks. For instance, Catalyst Center uses VXLAN in the data plane, and VXLAN encapsulation adds 50 bytes to the original packet. This can cause MTU issues, especially for applications with additional overhead, such as wireless. To prevent fragmentation, it is advisable to increase the MTU to 9100 bytes on interfaces across switches and routers in the fabric. This consideration becomes even more critical in WAN environments where your ISP mandates a 1500-byte MTU. In such cases, adjusting the MTU at the source ensures proper network functioning.

Addressing Security Considerations in SDN

A centralized SDN controller simplifies security management by providing a single point of control for configuring and enforcing security policies across the network. However, it is essential to ensure the controller itself is secured. The centralized nature of SDN introduces the risk of controller attacks, which can disrupt or manipulate the entire network if compromised.

Man-in-the-middle attacks can also compromise or spoof an SDN controller. An unauthorized access to the SDN controller, management interfaces, or network devices can enable attackers to gain control of the network or access sensitive data. This includes unauthorized logins to physical resources and access to configuration files or binaries. Insecure communication channels between SDN components can expose sensitive information and allow attackers to intercept or modify control traffic, affecting communication between the controller and switches as well as between different controllers in a distributed architecture.

To mitigate these risks, you can implement a comprehensive set of security measures. Start with controller hardening by ensuring that your SDN controller is updated with the latest security patches and configured according to best practices. Implement strong authentication and authorization mechanisms to control access. This includes using multifactor authentication (MFA) and granular role-based access control (RBAC) to define specific privileges for different users and roles.

Catalyst Center and SD-WAN controllers offer robust RBAC, allowing for precise control over who can perform specific actions within the network. This ensures that only authorized personnel can make critical changes, reducing the risk of unauthorized access or accidental misconfigurations.

Encryption is another critical component. Ensure that all control plane and data plane traffic between the SDN controller and network devices is encrypted. Cisco Catalyst Center and SD-WAN controllers use encrypted SSH or TLS tunnels for communication with network devices, providing a secure channel that protects against eavesdropping and tampering.

Network segmentation is also essential. Segment your network to isolate critical components and limit the potential impact of a security breach. By dividing the network into distinct segments, you can contain threats and prevent them from spreading across the entire network.

By implementing these comprehensive security measures—controller hardening, strong authentication and authorization, encryption, and network segmentation—you can significantly reduce the risks associated with a centralized SDN controller and ensure a more secure and resilient network environment.

Planning for Multidomain and Cloud Integration

Plan for multicontroller SDN deployments to support distributed networks or large-scale environments. Cloud-based SDN controllers, such as Cisco Meraki Dashboard and SD-WAN controllers, offer scalability, elasticity, and simplified management. They leverage cloud infrastructure for the SDN control plane, reducing on-premises infrastructure requirements and simplifying deployment. On-premises controllers like Cisco Catalyst Manager and the data center APIC controller should also aspire to key best practices:

- **Scalability:** Multicontroller architectures enhance scalability by distributing the control plane load across multiple controllers. This allows for the management of larger networks and a higher number of devices without performance degradation.

- **Resilience:** Deploying controllers in different regions or availability zones increases network resilience. If one controller fails, others can continue to operate, minimizing downtime and ensuring business continuity.

- **Reduced Latency:** Placing controllers closer to the network edges reduces latency for control traffic, improving responsiveness and overall network performance. This is crucial for applications requiring real-time communication or low latency, such as financial trading or industrial automation.

- **Support for Disjoint Transports:** Multicontroller deployments enable support for disjoint transport providers, allowing different network segments to use separate providers or technologies. This lets organizations leverage cost-effective solutions without sacrificing connectivity and performance.

By carefully considering these factors and planning for seamless interoperability between different domains and cloud environments, organizations can leverage the full potential of SDN to create agile, efficient, and secure networks.

Stage 2: Deployment and Migration

Deployment and migration are crucial phases in the SDN lifecycle, involving the implementation and transition of the network to the new SDN architecture. The following sections will explore these stages, highlighting key considerations and processes.

Preparing the Network Infrastructure

Before you deploy an SDN solution, it is crucial to assess the compatibility of the existing network infrastructure. This includes

- **Hardware Compatibility:** Determine if the current network devices, such as switches, routers, and wireless controllers, support the required SDN features.
- **Software Compatibility:** Verify that the software images running on the network devices are compatible with the chosen SDN controller and support the necessary protocols and functionalities.

For instance, when deploying Cisco SD-Access, you need to check if the switches meet the hardware requirements, including legacy models from the Catalyst 3850/3650, 4500E, and 6500/6800 series, as well as Nexus 7700 switches if they are in the existing network. Routing platforms should have ample DRAM (at least 8 GB, preferably 16 GB or more) to accommodate registered prefixes for the entire fabric domain. Additionally, ensure that the wireless infrastructure, including controllers and access points, is compatible. For SD-WAN, it is essential to verify whether the existing routers can run the Cisco IOS-XE SD-WAN image and support required features like VRF-aware IPsec.

Once compatible hardware and software are confirmed, configure the network devices with the necessary settings for SDN operation. This involves

- **Installing Correct Software Images:** Ensure all devices are running the required SDN-compatible software images. This task might involve downloading and installing new images or activating pre-installed images.

- **Configuring SDN-Specific Settings:** Configure SDN-specific settings on each device, such as enabling SDN protocols, defining controller connectivity parameters, and setting up security features.
- **Testing and Validation:** After configuration, thoroughly test each device to ensure it can communicate with the SDN controller and function correctly within the SDN environment.

For example, when you're deploying an SD-WAN-only site, it is necessary to install the Cisco IOS-XE SD-WAN image on the target router. Additionally, configure settings like VRFs, IPsec tunnels, routing protocols, and NAT to enable proper communication and traffic forwarding. Similar configuration steps are required for SD-Access deployments, including configuring switches for fabric participation, enabling features like VXLAN, and setting up connectivity to the Catalyst Center.

Maintaining accurate and up-to-date network documentation is essential for a smooth SDN transition. This documentation should include

- **Device Inventories:** A comprehensive list of all network devices, their models, serial numbers, software versions, and locations.
- **IP Addressing Schemas:** Detailed information on IP address allocation, including subnets, VLANs, and any reserved IP addresses for specific purposes like SDN controllers or management interfaces.
- **VLAN Configurations:** A clear representation of the VLAN structure, including VLAN IDs, names, and their purpose within the network.
- **Underlay Routing Protocol:** The underlay network provides the physical connectivity for the SDN overlay and relies on a robust routing protocol. Choose a routing protocol that your operational team is well-versed in and that exhibits the scalability to accommodate the size and complexity of the network. Additionally, fast convergence times are crucial for maintaining network stability and minimizing downtime during changes.

This detailed documentation aids in troubleshooting, planning future upgrades, and ensuring consistent configuration across the network. It also provides a valuable resource for security audits and compliance reporting. For example, when you're deploying SD-Access, having a detailed IP address scheme for different VNs, as well as information on DHCP and DNS server assignments for each VN, is crucial for proper network operation. Maintaining accurate documentation for VLAN configurations, especially when integrating wireless networks into the SD-Access fabric, is essential for seamless connectivity and security policy enforcement.

Deploying the SDN Controller

This section shows the process of deploying the SDN controller, a core component of any SDN architecture. This process involves installing, configuring, and securing the controller for optimal performance and reliability.

Installing and Configuring the SDN Controller

The SDN controller can be deployed in various forms:

- **Physical Appliance:** Provides dedicated hardware for robust performance and security.
- **Virtual Machine:** Offers flexibility in deployment and resource allocation within virtualized environments either on-prem or cloud hosted.
- **SaaS Controller:** Provides a scalable and managed solution, leveraging the vendor's expertise to host and maintain controller infrastructure while providing a guaranteed SLA for availability.

Installation parameters of the SDN controllers are as follows:

- **Physical Appliance:** Involves physically installing the appliance in a secure data center rack, connecting it to the network and power, and booting up the system.
- **Virtual Machine:** Requires creating a virtual machine on a suitable hypervisor platform, configuring its resources (CPU, memory, disk space), and installing the controller software image.

- **SaaS Controllers:** Involves subscribing to the SDN controller service from a cloud provider, configuring service parameters, and deploying the controller instance in the chosen cloud region.

Configuration

Once installed, the controller needs to be configured based on the specific SDN solution and network requirements. Common configuration tasks include

- **Network Settings:** Assigning IP addresses, subnet masks, and default gateways to the controller's management and data interfaces.
- **Controller-Specific Settings:** Configuring parameters specific to the SDN controller, such as domain names, authentication settings, logging levels, and other operational parameters.
- **Integration with External Systems:** Integrating the controller with other systems like authentication servers (e.g., RADIUS, TACACS+), monitoring tools, and network management systems.
- **Policy Configuration:** Defining network policies that will govern traffic forwarding, security, and quality of service (QoS) within the SDN environment.

For example, in a Cisco SD-Access deployment, the Catalyst Center serves as the SDN controller. Initial setup involves choosing the appropriate hardware platform, potentially a DN3-HW-APL-XL appliance based on the Cisco UCS C240 M6 platform for larger deployments, or a smaller platform for less-demanding environments. Configuration encompasses tasks like setting up network connectivity, integrating with ISE for authentication and policy enforcement, and establishing BGP peering for route distribution.

In an SD-WAN deployment, the Catalyst Manager platform acts as the controller, requiring configuration for tasks such as defining network templates, provisioning edge devices, and setting up control and data plane policies. It also involves integrating Catalyst Manager with other SDN components like Catalyst Controller (for routing decisions) and Catalyst Analytics (for monitoring and troubleshooting).

Setting Up High Availability and Disaster Recovery

Ensuring continuous operation of the SDN controller is crucial for uninterrupted network services. This involves implementing

- **High Availability (HA):** Deploying redundant controllers in an active-passive or active-active configuration to eliminate single points of failure. If the active controller fails, a standby controller takes over seamlessly, minimizing downtime.

- **Disaster Recovery (DR):** Replicating the controller and its configuration data to a geographically separate location. In case of a disaster impacting the primary data center, the controller operations can be restored at the DR site, ensuring business continuity.

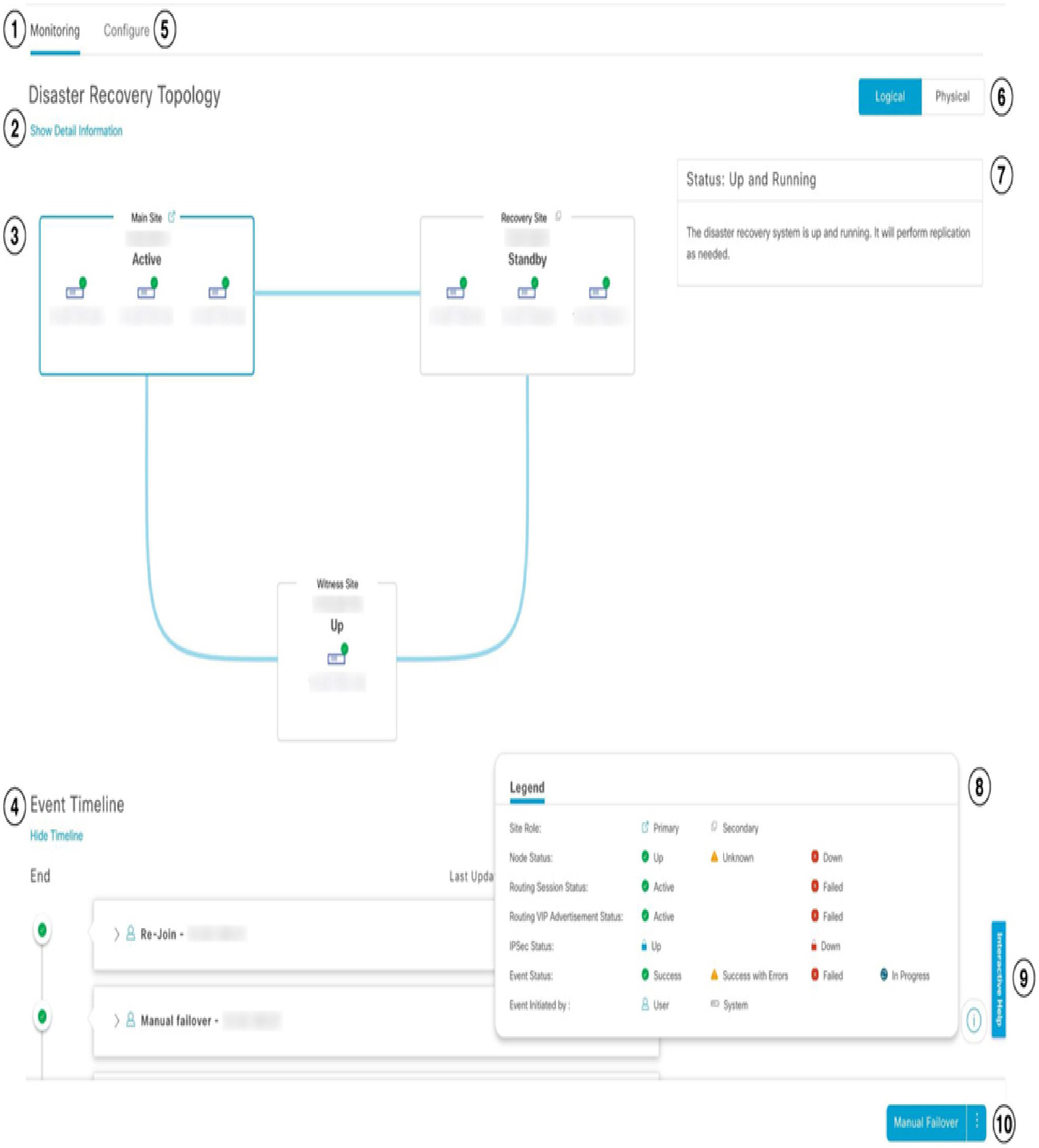
Specific mechanisms for HA and DR vary depending on the SDN solution and the chosen deployment model (physical, virtual, or cloud-based). Common techniques include

- **Clustering:** Grouping multiple controller instances into a cluster, where they share the workload and provide redundancy.

- **Data Replication:** Replicating controller configuration and state data to a standby or DR site.

- **Geographic Redundancy:** Deploying controllers in geographically diverse locations to mitigate risks associated with regional outages.

For instance, the Cisco Catalyst Center can be deployed in an HA configuration, forming a cluster for redundancy. In a multiregion deployment, multiple Catalyst Center clusters can be deployed, providing regional resilience. A similar approach can be adopted for the Catalyst Manager platform in SD-WAN deployments, with multiple Catalyst Manager nodes forming a cluster for HA and geographically separated clusters for DR. Figure 12-3 shows an example for Catalyst Center's Disaster Recovery Solution.



Status: Up and Running

The disaster recovery system is up and running. It will perform replication as needed.

Figure 12-3 *Catalyst Center's Disaster Recovery Solution*

Prioritizing Network Services for SDN Migration

When you're migrating network services to SDN, it is essential to prioritize them based on their suitability for SDN and the potential benefits they can reap. This approach ensures a smoother transition and maximizes the value derived from the SDN implementation. A phased migration plan is crucial to minimize disruptions and ensure service continuity. This approach involves breaking down the migration into manageable stages, allowing for thorough testing and validation at each step.

- **Initial Assessment:** Conduct a comprehensive assessment of the existing network infrastructure, including device inventory, network topology, and current service configurations. This information forms the basis for the migration plan.
- **Pilot Phase:** Start with a small-scale pilot migration, targeting a noncritical area of the network. This allows for testing the SDN solution in a real-world environment, identifying potential issues, and refining the migration process before wider deployment. Services prioritized in the previous step can be included in this pilot phase.
- **Incremental Migration:** Based on the pilot's success, gradually expand the SDN deployment to other network segments. This incremental rollout minimizes the impact on user experience and addresses any unforeseen challenges as they arise.
- **Service Migration:** Migrate network services according to the prioritization matrix, starting with high-priority services. Ensure comprehensive testing and validation before moving each service to the SDN environment.
- **Legacy Integration:** Address the integration of legacy network devices and services with the SDN architecture during the migration. This step may involve using techniques such as tunneling, protocol translation, or hybrid approaches to bridge the gap between traditional and SDN environments.

Testing and Validating Migration Success

Implementing appropriate testing and validation procedures at each stage of the migration is crucial to guarantee a successful and reliable transition.

- **Functional Testing:** Verify the functionality of network services after they are migrated to the SDN environment. This includes validating connectivity, routing, policy enforcement, and application performance.

- **Performance Testing:** Assess the performance of the SDN solution under different traffic loads and network conditions. Measure key metrics such as latency, throughput, and packet loss to ensure the SDN infrastructure meets performance requirements.
- **Security Testing:** Evaluate the security posture of the SDN environment by validating authentication mechanisms, access control lists (ACLs), and firewall rules. Conduct vulnerability assessments and penetration tests to identify and mitigate potential security weaknesses.
- **Rollback Planning:** Develop a rollback plan to revert to the previous network configuration if any critical issues arise during the migration. This plan should outline the steps to restore services and configurations to their premigration state.

Stage 3: Operations and Management

Software-defined networking offers a transformative approach to network management, introducing centralized control and programmability, leading to enhanced agility, efficiency, and scalability. The effective management of an SDN environment requires a thorough understanding of its architecture and the implementation of the right practices to ensure smooth operations. This involves monitoring network performance, maintaining security protocols, and ensuring seamless integration with existing systems. Furthermore, organizations should adopt industry best practices to maximize the benefits of SDN. This effort may include training staff on SDN concepts, establishing strong governance frameworks, and continuously evaluating network performance to identify areas for improvement. By focusing on these operational aspects and committing to effective management practices, organizations can unlock the full range of advantages that SDN offers, as discussed in the following points.

Integrating SDN Automation with Existing IT Operations Management Systems

Integrating SDN automation with existing IT operations management systems is vital for gaining a unified view of network operations. This integration facilitates seamless information flow between different systems and streamlines IT processes, enhancing efficiency and network visibility.

- **IT Service Management (ITSM) Integration:** Automating the creation of service tickets in ITSM based on network events enables a more efficient incident management process. This integration can be achieved by configuring the SDN controller to send alerts or notifications to ITSM when specific events occur, such as device failures, performance degradation, or security breaches. This allows IT teams to respond to issues more quickly and effectively, reducing downtime and improving service quality.

- **Monitoring and Logging Systems:** Integrating SDN controllers with existing monitoring and logging systems provides a centralized view of network health and performance. This allows for proactive issue identification and resolution. By collecting and correlating data from various network devices and applications, IT teams can gain insights into network trends, identify potential bottlenecks, and proactively address performance issues before they impact users.

- **Security Information and Event Management (SIEM) Systems:** Integrating security policy enforcement with SIEM systems enhances threat detection and response capabilities. This is achieved by correlating network security events with other security data sources. By analyzing security logs from the SDN controllers, firewalls, intrusion detection systems, and other security tools, SIEM systems can identify and respond to security threats more effectively.

- **Achieving Seamless Integration:** APIs and integration tools play a crucial role in achieving seamless integration between SDN automation and existing IT workflows. SDN controllers typically expose APIs that allow other systems to interact with them programmatically. This enables the development of custom integrations tailored to the specific needs of an organization. For example, an organization could develop a script that automatically provisions new users in their identity management system and simultaneously configures the necessary network access policies on the SDN controller. This ensures that new users have immediate access to the network resources they need without manual intervention.

Monitoring and Troubleshooting the SDN Environment

Effective monitoring and troubleshooting strategies are essential in SDN environments to maintain network health, performance, and security. Various tools and techniques can proactively identify and resolve issues within the SDN ecosystem.

Implementing Comprehensive Monitoring Tools

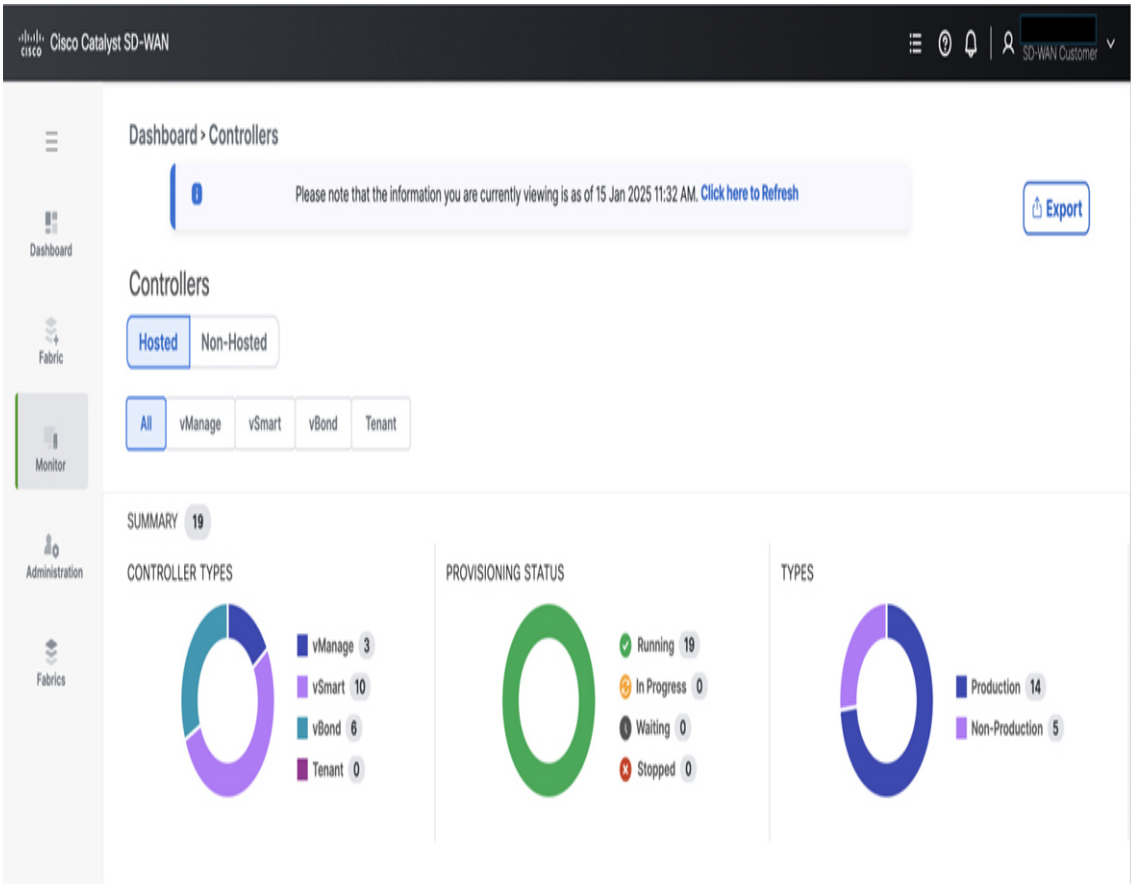
Comprehensive monitoring tools are crucial for tracking various aspects of the SDN environment, including network performance, traffic flows, security events, and controller health.

- **Network Performance Monitoring:** Tools like SNMP and Flexible NetFlow, enhanced within Cisco Catalyst Center, provide statistics on user groups, traffic flows, and device performance metrics. This data enables network administrators to identify performance bottlenecks, track application usage, and optimize network resources.

- **Traffic Flow Analysis:** Analyzing network traffic flows using tools like NetFlow helps you understand network behavior, identify anomalous traffic patterns, and diagnose connectivity issues. Visualizing traffic flows within the SDN controller's dashboard provides insights into how traffic traverses the network and can highlight potential areas of concern.

- **Security Event Monitoring:** Integrating SDN controllers with security information and event management (SIEM) systems allows for comprehensive monitoring of security events. Correlating data from various security sources, such as firewalls, intrusion detection systems, and the SDN controller, enhances the detection and response to security threats.

- **Controller Health Monitoring:** Ensuring the health and availability of the SDN controller is critical for maintaining network stability. Platforms like Catalyst Manager, the central management tool for Cisco SD-WAN, provide dashboards for monitoring the health and status of the SD-WAN overlay network, including controller connectivity, device status, and alarm notifications (see Figure 12-4).



Please note that the information you are currently viewing is as of 15 Jan 2025 11:32 AM. Click here to Refresh

Figure 12-4 *Controller Health Monitoring*

Utilizing SDN Controller's Visibility Features

SDN controllers offer enhanced visibility features that provide insights into network behavior and aid in identifying potential issues.

- **Centralized Dashboards:** SDN controllers provide centralized dashboards that offer a comprehensive view of the network (see

Figure 12-5). These dashboards display real-time information on device status, network performance, traffic flows, security events, and controller health. This centralization simplifies network monitoring and enables quicker identification of potential issues.

- **Topology Visualization:** Visualizing the network topology within the SDN controller provides a graphical representation of network devices and their interconnections. This view helps you understand traffic paths, identify network segmentation boundaries, and pinpoint areas affected by failures or performance degradation.
- **Real-Time and Historical Data:** SDN controllers collect and store both real-time and historical network data. Analyzing historical data trends helps you identify recurring issues, understand network usage patterns, and plan for capacity upgrades. Real-time data enables quick identification of network anomalies and facilitates rapid response to critical events.
- **Drill-Down Capabilities:** SDN controllers offer drill-down capabilities that allow network administrators to investigate specific devices, flows, or events in greater detail. This feature enables in-depth analysis of network behavior and helps pinpoint the root cause of issues more effectively.

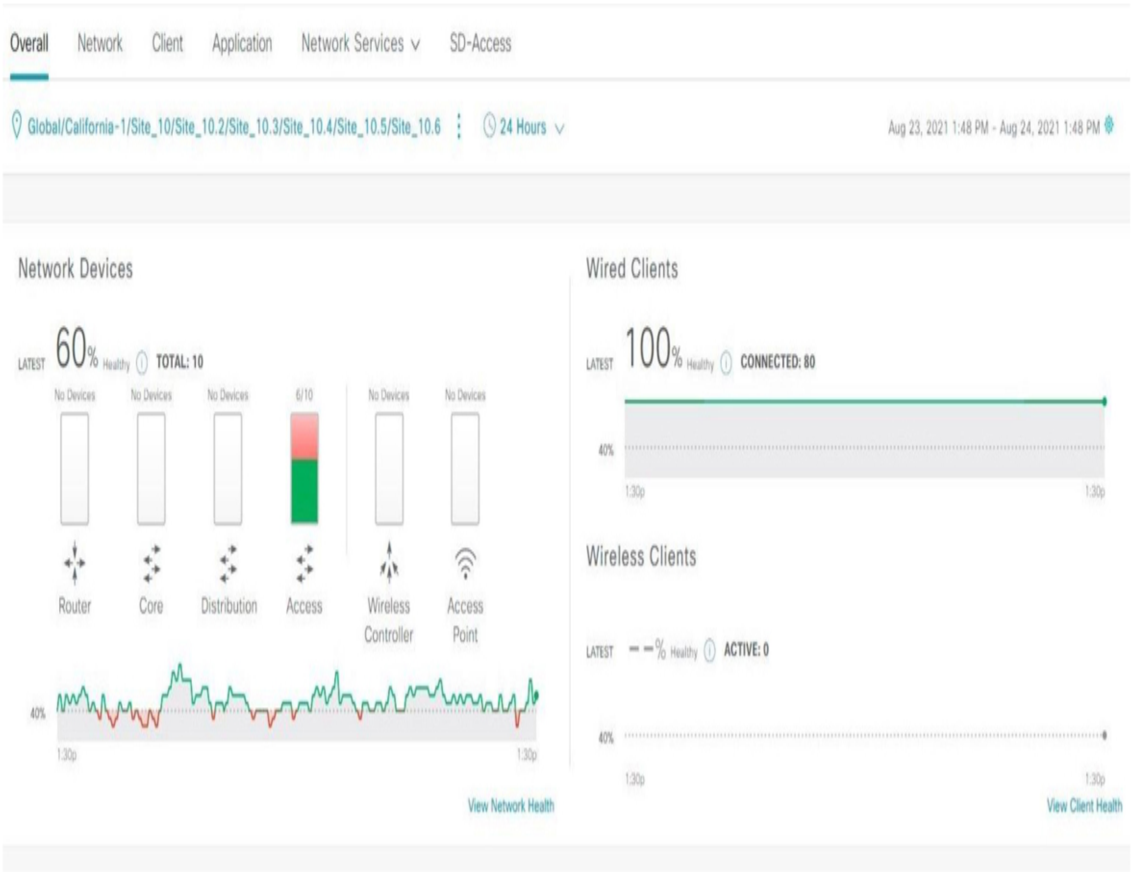


Figure 12-5 *SDN Centralized View of Network and Health*

Developing Troubleshooting Procedures

Developing troubleshooting procedures specific to the SDN environment is essential for addressing the unique challenges posed by the centralized control and software-defined nature of SDN.

- **Understanding the SDN Architecture:** It is crucial to have a thorough understanding of the SDN architecture, including the roles and interactions between the controller, network devices, and applications. This knowledge helps isolate issues to specific components of the SDN ecosystem.
- **Recognizing Control and Data Plane Separation:** Recognizing the separation of control and data planes in SDN is vital for troubleshooting. Issues in the control plane can affect overall network behavior, while data plane issues typically impact specific flows or devices.

- **Leveraging Controller Logs and Events:** SDN controllers maintain logs and event records that provide valuable information for troubleshooting. Analyzing controller logs can reveal configuration errors, communication failures, and other events that might have contributed to the issue.
- **Verifying Network Device Configuration:** Ensuring that network devices are correctly configured and communicating with the controller is essential. Tools within the SDN controller allow for the verification of device configuration, status, and connectivity.
- **Analyzing Application Behavior:** Understanding how applications interact with the network is crucial for troubleshooting application-specific issues. Tools like NetFlow can provide insights into application traffic patterns, helping to identify potential bottlenecks or misconfigurations.
- **Establishing Collaboration and Communication:** Effective troubleshooting often requires collaboration between network, security, and application teams. Establishing clear communication channels and sharing relevant information between teams can expedite issue resolution.

By implementing comprehensive monitoring tools, utilizing SDN controller visibility features, and developing specific troubleshooting procedures, organizations can proactively manage and maintain their SDN environments, ensuring network reliability, performance, and security.

Maintaining Security and Compliance

Regular security audits and vulnerability assessments are essential for identifying and mitigating potential risks in the SDN environment. These assessments help uncover weaknesses and provide recommendations for improvement. Establishing a network service directory with access control lists is a best practice. This directory acts as a centralized repository for information about network services, enabling the SDN controller to control access and prevent unauthorized use.

Recommended Security Measures

Several measures are recommended to maintain security and compliance in an SDN environment:

- **Implement Access Control Mechanisms:** Implementing access control mechanisms is essential to restrict unauthorized access to the SDN controller and network infrastructure. Role-based access control is a widely adopted method for managing user permissions in SDN environments. RBAC allows administrators to assign specific roles to users and grant permissions based on those roles, preventing unauthorized individuals from performing sensitive actions and ensuring users have access only to the resources they need.

- **Use Strong Authentication Mechanisms:** Implement strong authentication for all network nodes, including multifactor authentication where possible. Multifactor authentication adds an extra layer of security by requiring multiple forms of identification before granting access.

- **Protect SDN-to-Device Communication:** Enable encryption and integrity protection for southbound interfaces to prevent rogue devices from providing false information to the SDN controller. This ensures that only authorized devices can communicate with the controller and that the communication is secure.

- **Maintain Configuration Compliance:** Network configurations are results of the policies defined on SDN interface. Regular validation helps the configuration to detect any out-of-band changes that may indicate an attack or misconfiguration.

- **Encrypt the SDN Database:** Encrypt persistent information stored within the SDN environment to protect sensitive data from unauthorized access. Encryption ensures that even if attackers gain access to storage systems, they cannot read the sensitive information.

- **Restrict Physical Access:** Restrict physical access to SDN resources, including servers and virtual machines in data centers. This helps prevent unauthorized individuals from tampering with the SDN infrastructure.

- **Keep Systems Updated:** Keeping the SDN controller and network devices updated with the latest security patches and firmware releases is crucial for addressing newly discovered vulnerabilities. Software updates often include patches for security flaws that could be exploited by attackers. Regular updates reduce exposure to these threats.
- **Ensure Compliance with Industry Standards:** Compliance with relevant industry standards and regulations is crucial for organizations operating in regulated industries. Standards like NIST SP 800-207, which outlines a zero trust architecture, provide guidance on best practices for securing modern networks. Adhering to these standards demonstrates a commitment to security and reduces the risk of legal or financial penalties.

By implementing these comprehensive security measures and maintaining compliance with industry standards, organizations can ensure the security and reliability of their SDN environments.

Summary

This chapter offered valuable insights into the essential considerations and best practices for effectively deploying software-defined networking (SDN). It emphasized the importance of comprehensive planning, design, implementation, and ongoing management to maximize the benefits of SDN. By adopting SDN solutions, organizations can enhance network flexibility, scalability, and security. Start by clearly defining your objectives and assessing your network's readiness. Focus on scalability and prioritize network security throughout the implementation process. Investing in training and skill development for your IT team will empower them to navigate the new SDN landscape successfully. Before launching your SDN solution in a production environment, conduct thorough testing and validation to ensure reliability. Don't hesitate to seek support from vendors or outside consultants because cutting-edge information may only be available from specialists. Additionally, establishing a robust change management plan will facilitate a smooth transition. Ongoing monitoring and optimization of your SDN infrastructure are crucial for achieving and maintaining optimal performance. With thoughtful planning and adherence to these best practices, you can unlock the full potential of SDN solutions and significantly enhance your network operations.

References

1. The PPDIOO Model: https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/catalyst-center/2-3-7/admin_guide/b_cisco_catalyst_center_admin_guide_237/b_cisco_dna_center_admin_guide_2_3_7_chapter_0111.xhtml
2. Best Practices: https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/dna-center/hardening_guide/b_cisco_catalyst_center_security_best_practices_guide.xhtml
3. Cisco Enterprise Campus Architecture: https://www.ciscopress.com/articles/article.asp?p=1608131&seqNum=3
4. Compatibility Matrix: https://www.cisco.com/c/dam/en/us/td/docs/Website/enterprise/catalyst_center_compatibility_matrix/index.xhtml
5. User Guide: https://www.cisco.com/c/en/us/td/docs/cloud-systems-management/network-automation-and-management/catalyst-center-assurance/2-3-7/b_cisco_catalyst_assurance_2_3_7_ug/b_cisco_catalyst_assurance_2_3_6_ug_chapter_01.xhtml

Chapter 13. Wired and Wireless Assurance

In this chapter, you will learn about the following:

- Wired best practice design concepts
- Wireless best practice design concepts
- Wireless deployment constructs
- Anchoring concepts (Meraki, Catalyst)
- TrustSec monitoring and security enforcement
- Case studies from the field

What Is the Best Practice for Your Enterprise Architecture?

Many customers come to Cisco Professional Services teams and our partners to identify how to deploy a best practice design for their architecture and network, in ever-evolving and changing IT landscapes. Over the years, technology requirements have changed in the form of increased needs in bandwidth, which has fueled better-quality network connectivity from service providers, to increased mobility. Enterprise environments are moving from fixed workplaces to open-plan dynamic workplaces within many corporate offices and environments. As much as these changes have fostered an increase in speed, stability, and scale, security has also become a key tenet of modern network architectures.

Legacy networks, as simplistic and modern as they may have been in their heyday, would be considered extremely insecure, to the point that today many security leaders would consider them negligent. Thankfully, over the years, the industry has come a long way, incrementally, by simply increasing bandwidth, standardizing protocols, and improving encryption methods.

For many years, Cisco has been publishing *Cisco Validated Designs (CVDs)*, which provide tried-and-true architectural representations of how a good state should look for a particular technology. Does a CVD match every single use case and every single scenario from all of our customers globally? No, it does not. However, it does attempt to get to a point where most customer architectures and designs, particularly in the small to medium business space, can be realized.

For the networks and architectures that do not match up directly to a CVD, the documentation itself should not be disregarded; it should still be considered as a strong bearing to orient toward in building a solid network design.

Wired Network Best Practice Design Concepts

Wired Ethernet-based networks were the starting point for networks deployed over the years and still remain a key tenet for deployment in critical environments today. It remains difficult to compare the reliability and robustness of wired architectures with any radio frequency (RF)–based technology that may be available on the market today. While manufacturing environments are slowly shifting across to RF-based technologies, such as private 5G in some cases, wired Ethernet-based deployments continue to remain the norm in many critical environments today.

To ensure that the right standards are maintained to attain a scalable, robust, and resilient architecture that can achieve modern zero trust security approaches, a solid foundation is necessary. It must map the criticality of services to the corresponding network structures, bandwidth demands, and hardware constructs to support an architecture that avoids service impact or loss of network traffic during hardware or software upgrades.

When an organization is building a campus-switching architecture, it is important to consider scale and oversubscription ratios. Using ratios that allow for the target throughput in the architecture is key in ensuring that the deployment can handle the expected throughput from the network architecture. In terms of traffic ratios between network layers, the general rule for deployment has largely remained the same over the years. Advancements are being made in the speed and capabilities of devices connecting to the access layer, such as access points (APs) being capable of achieving multiple gigabytes of throughput. While the ratios shown in Figure 13-1 continue to provide a good orientation, in scenarios where heavy bandwidth usage is required at the access layer, an organization may need to consider an increase in capacity.

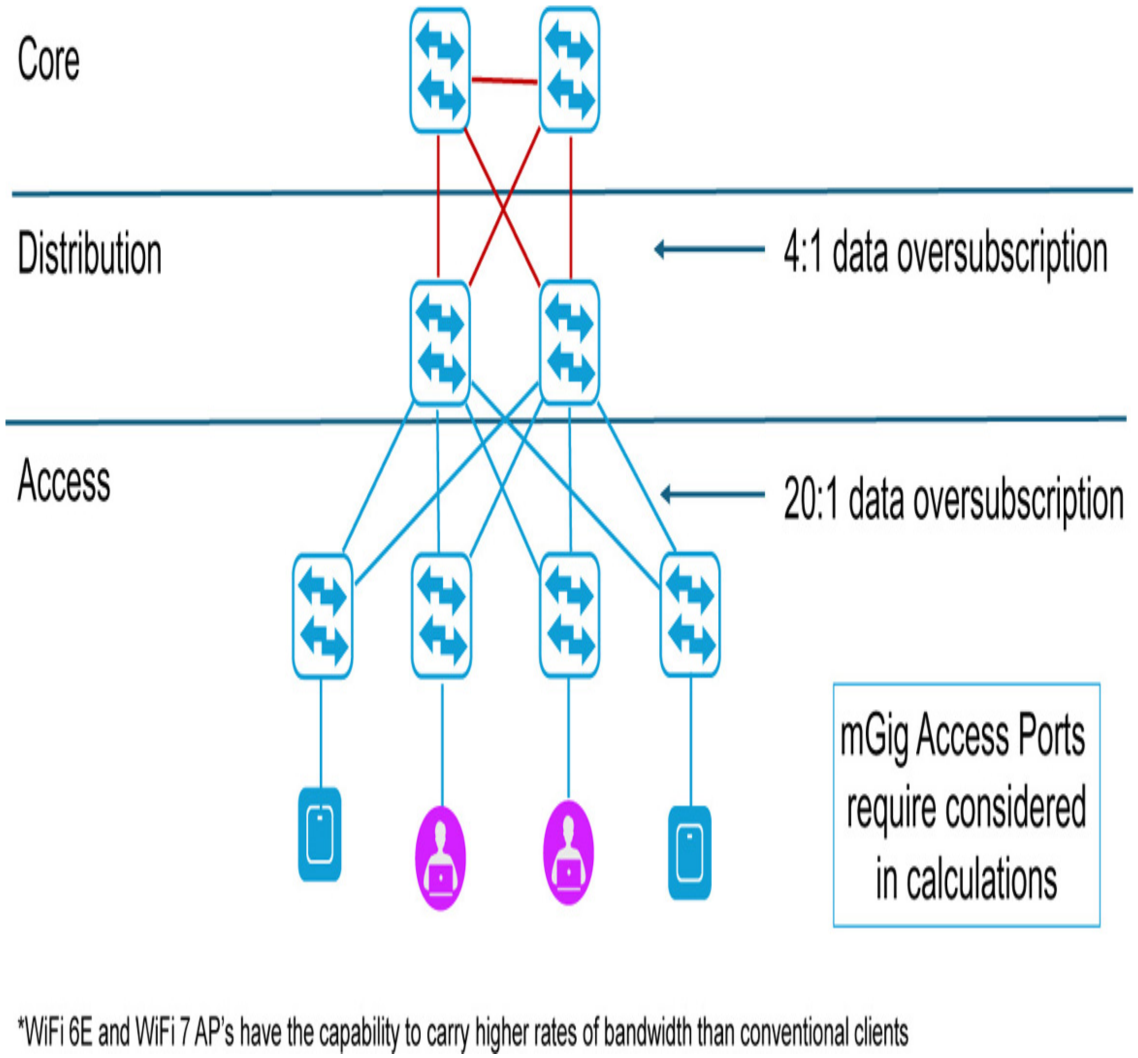

Core
Distribution
4:1 data oversubscription
Access
20:1 data oversubscription
mGig Access Ports require considered in calculations
*WiFi 6E and WiFi 7 AP's have the capability to carry higher rates of bandwidth than conventional clients

Figure 13–1 *Oversubscription Ratios*

When considering options for the buildup of a network, an organization can choose various paths toward the deployment of physical wired architectures. The first question that needs to be considered is whether the target architecture should be Layer 2 or Layer 3. Over the past few decades, classic architectures and designs have largely been focused around building up hierarchical structures using Layer 2, which relies on the Spanning Tree Protocol (STP) from the distribution layer of the network down to the access layer.

Note

You can find a more comprehensive overview of Spanning Tree Protocol in Chapter 6, "Automating the Campus."

Such an architecture is often referred to as a *hierarchical* or *tiered network design*, as can be seen in Figure 13-2.

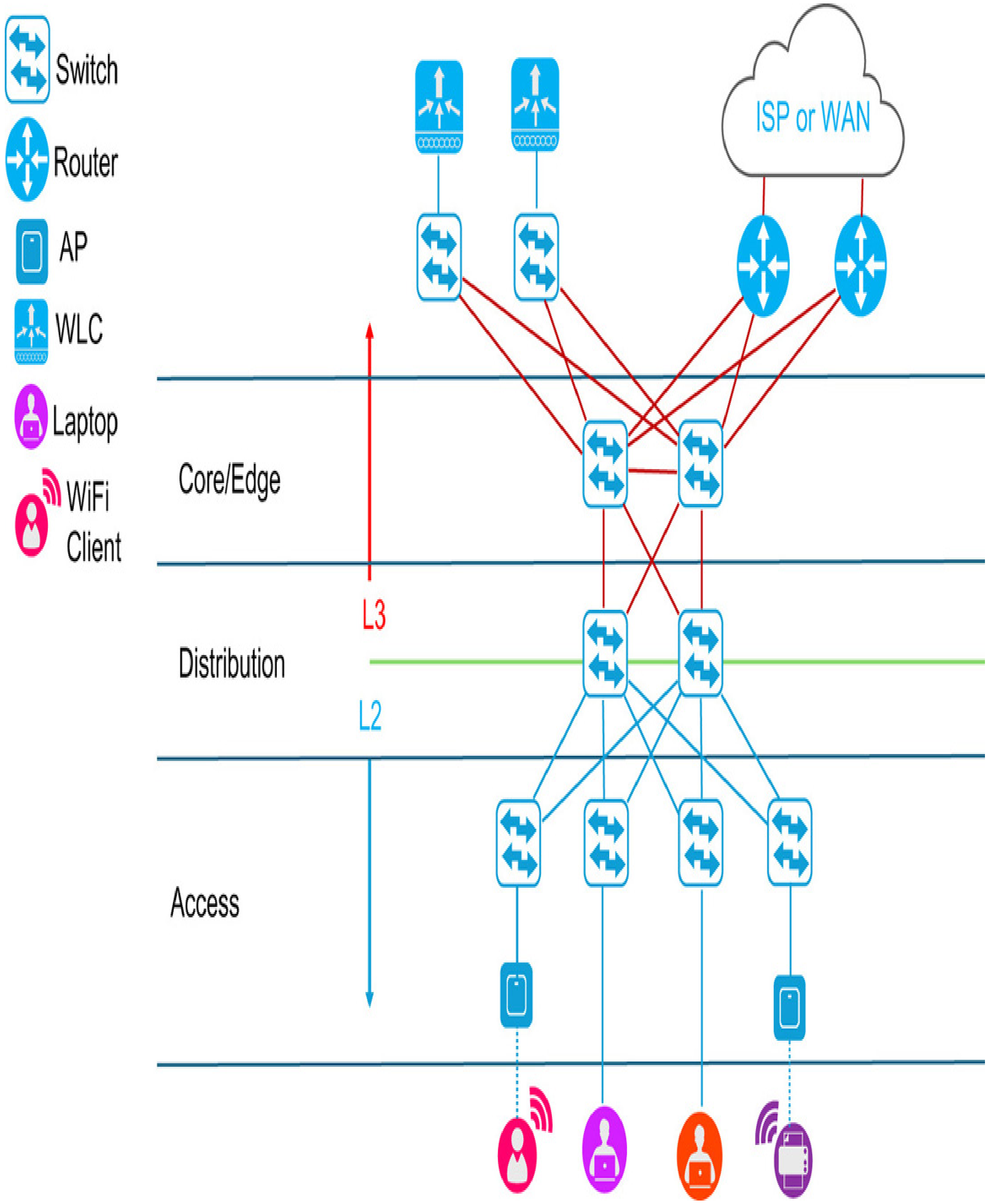


Figure 13-2 *Tiered Network Architecture*

Tiered Network Design

Tiered network designs enable organizations to provide a scalable architecture, setting a viable blueprint for a network that could support an increase in size and capacity if the correctly sized core and distribution layers are chosen based on projected organizational growth. The physical cabling mediums that are used to connect the network components and clients to the network are often selected based on the physical distance from component to component. Unshielded twisted pair (UTP) copper-based Ethernet connections are typically limited to ~100 meters (~328 feet) in distance. Other considerations in the context of cabling selection in highly sensitive environments such as defense networks include TEMPEST-rated equipment together with the use of fiber optic–based switching equipment. Such equipment selection can lower the likelihood of exfiltration via RF-based side channel attacks based on variations of *van Eck phreaking* techniques on compromised network systems or endpoints.

To aid in the creation of a resilient and quick-to-converge network, as can be seen in the architecture in the right two diagrams in Figure 13-4, the redundant links between each layer from distribution to access are always built up using port channels, binding redundant links together. The distribution layer switches themselves are also consolidated into switch stacks, using Stackwise virtual or physical stacking. The reason for this approach is to mitigate the use of a Spanning Tree Protocol enforcement mechanism that will result in the blocking of redundant links.

Figure 13-3 shows the functionality of Spanning Tree, where redundant links are blocked to avoid Layer 2 loop scenarios.

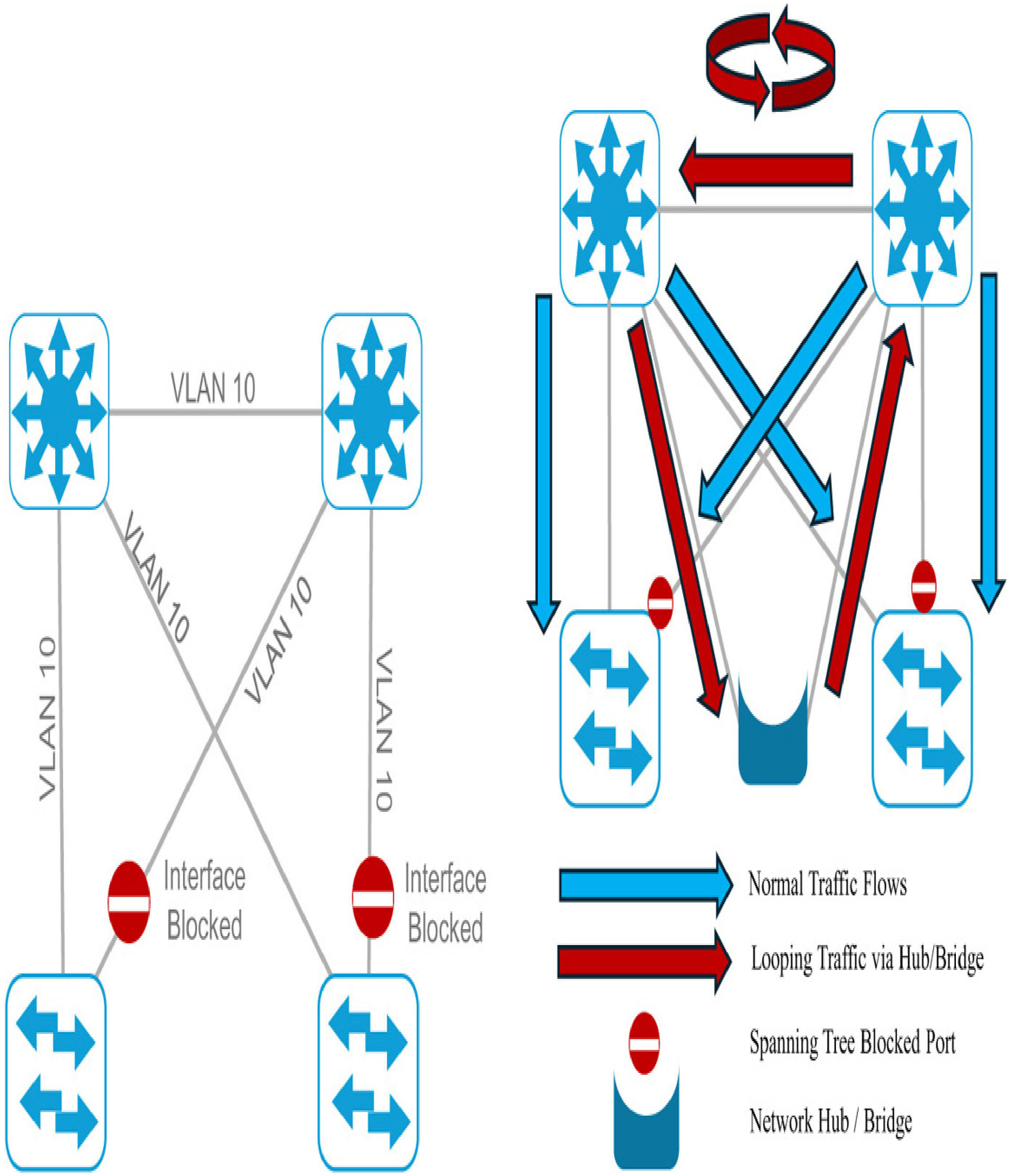


Figure 13-3 *Interfaces Blocked in Layer 2 Topology (Left); Layer 2 Loop Caused by Network Hub/Bridge (Right)*

This sort of architecture has served many networks well over the years and continues to be a common construct in various production environments. One of the key challenges that can appear in such a network architecture, however, is invariably the potential for a misconfiguration to take place that could result in a bridging loop.

While viable safeguards remain available to avoid negative scenarios that can trigger a bridging loop, the Cisco Technical Assistance Center (TAC) still receives regular calls from customers that have inadvertently had their network connect in a way that caused a loop scenario and usually, in conjunction with it, a significant outage of service. It is for this reason that the topology constructs that are detailed in Figure 13-4 (and mirrored in the CVD) are advisable for a healthy deployment.

In traditional Layer 2 architectures, a key limitation that, under certain circumstances, resulted in suboptimal forward paths was the Spanning Tree Protocol. To avoid bridging loops, as shown in Figure 13-3, switch-to-switch links would need to forward bridge protocol data units (BPDUs), which result in the selection of the links in a topology to block. While this technology provides many benefits around the avoidance of loops, it has the added caveat that, depending on the topology, certain links may not be adequately utilized. With strict active and standby forwarding paths (as depicted in Figure 13-3), it is even possible that 50 percent of the available bandwidth within a deployment may need to remain unused.

Of course, there are ways to mitigate this challenge. For example, it may be mitigated through the use of stacking, either through physical stacking using methods like Stackwise together with Etherchannels, or through virtual stacking using Stackwise Virtual together with Multi-chassis EtherChannel (MEC).

Figure 13-4 shows some examples of such topology constructs.

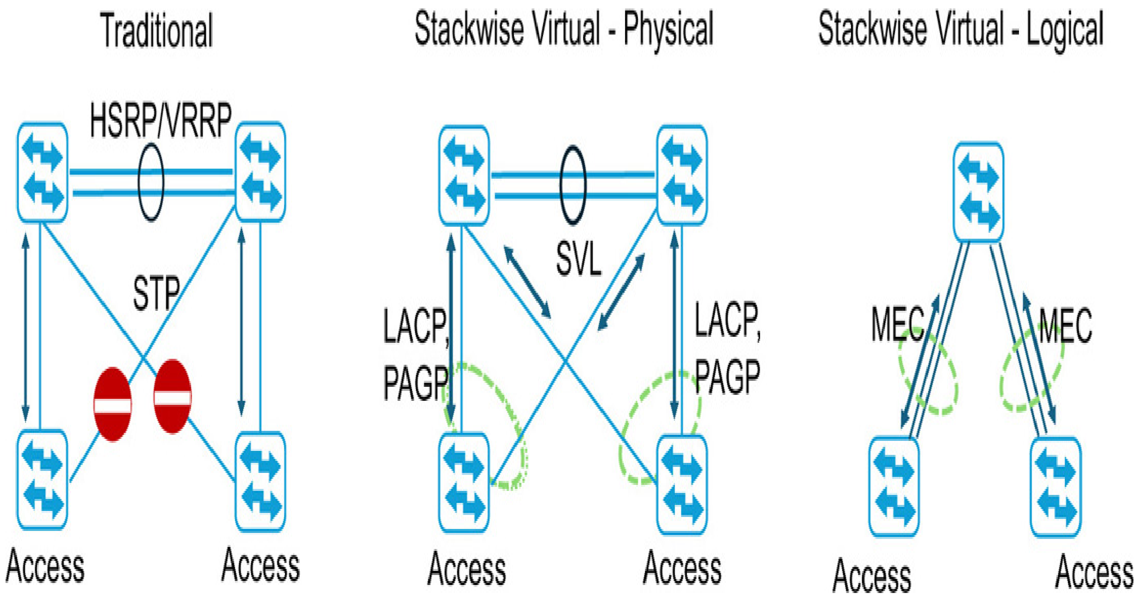


Figure 13-4 *Layer 2 Blocking Avoidance Topologies*

When you are deploying classic Layer 2 architectures rather than a Layer 3–based underlay, further security features are available for use, so as to avoid potential security or resilience impacts within the architecture. The following configurations should be considered for use in such constructs:

- **BPDU Guard:** Providing protection against rogue switches participating in Spanning Tree from being connected
- **Root Guard:** Prevention of newly introduced switches, assuming the SPT root switch role causing disruption
- **Loop Guard:** Mitigation of SPT forwarding state on ports withing received BPDUs

Stacking Constructs

Figure 13-4 shows both the physical and logical views of Stackwise Virtual. The advantage to this construct is that all links that are available to use can be used. The Stackwise Virtual pair is also seen as being one single switch despite there being a physical separation between the redundant devices. In Layer 2 networks, this solution provides many benefits over the traditional switching setup that is impacted by spanning tree convergence delays.

While there are some great benefits to this technology for individuals using Layer 2 architectures, it should be noted that some negatives also come with this topology. Although redundancy is achieved via port channels and the logical combination of switches, awarding redundancy on the data plane, redundancy is not present in the context of the control plane.

Looking at a simple example, let’s consider a collapsed core setup like the one shown in Figure 13-5. While the distribution switch in the stacked topology may be terminating two independent BGP peerings from the core layer, if there is an issue encountered in the BGP process on the distribution switch, the impact would affect all downstream devices.

In contrast, having separate distribution nodes that are active (and unstacked) can provide further redundancy from a control plane perspective as a result of having separate control plane processes running independently and not merged on a single system. This structure can provide added benefits in the context of routing plane resilience but also security control plane resilience for the redundancy of related protocols such as SXP.

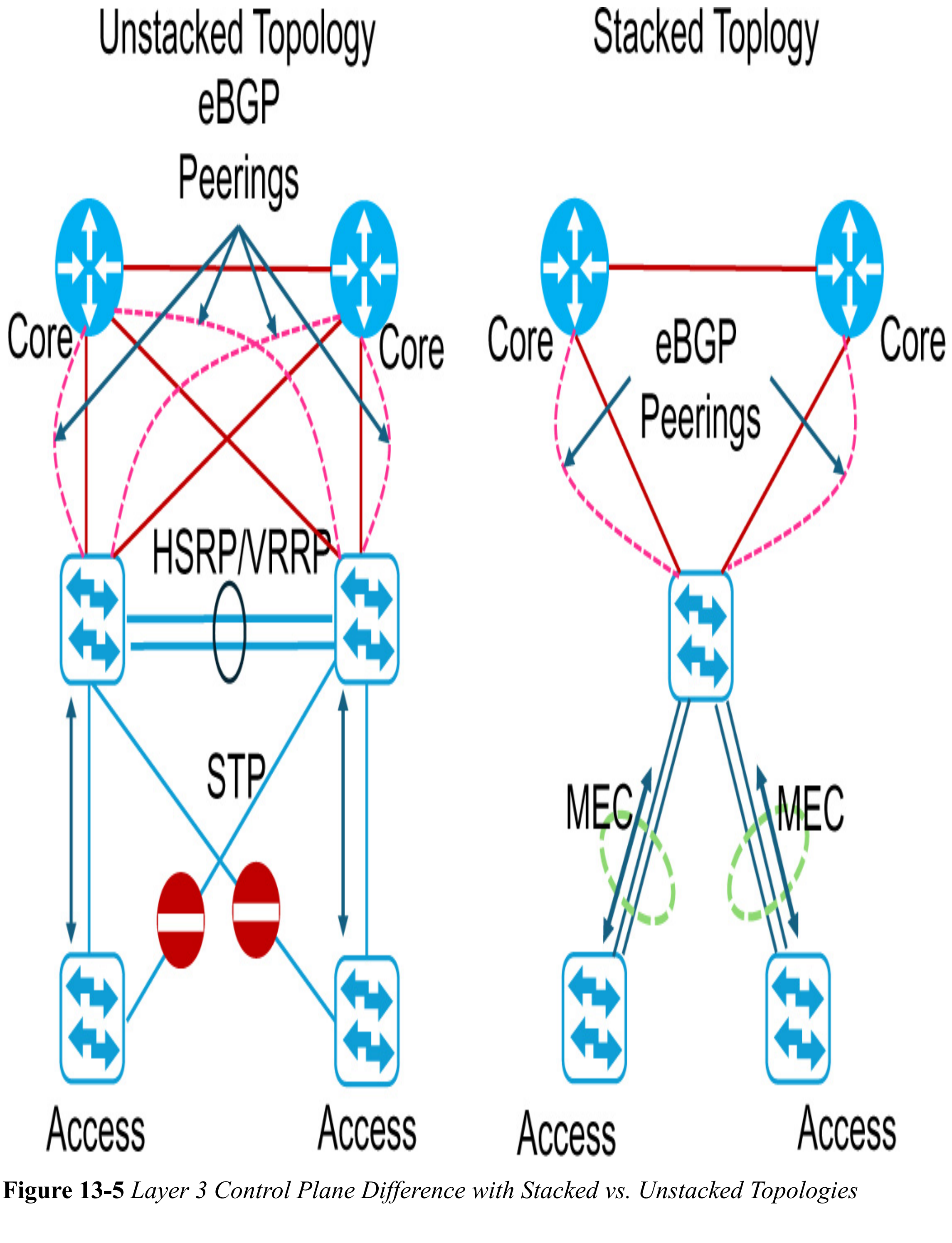


Figure 13-5 *Layer 3 Control Plane Difference with Stacked vs. Unstacked Topologies*

Layer 3 Architectures

Alternative design constructs that are focused on Layer 3 instead of Layer 2 have become much more popular over the years, providing network operators with a means to deploy architectures that are not at risk of Layer 2 bridging loops.

Three examples of popular Layer 3 constructs are shown in Figures 13-6; they include a Layer 3 routed design, an SD-Access Layer 3 fabric, and an EVPN Layer 3 fabric architecture.

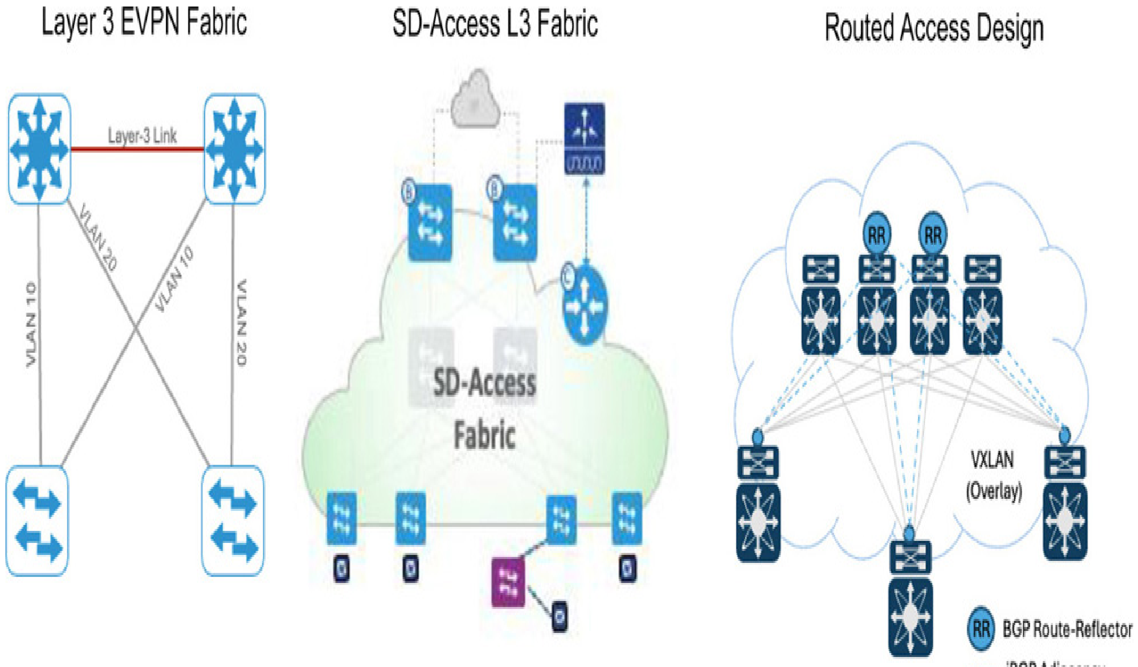
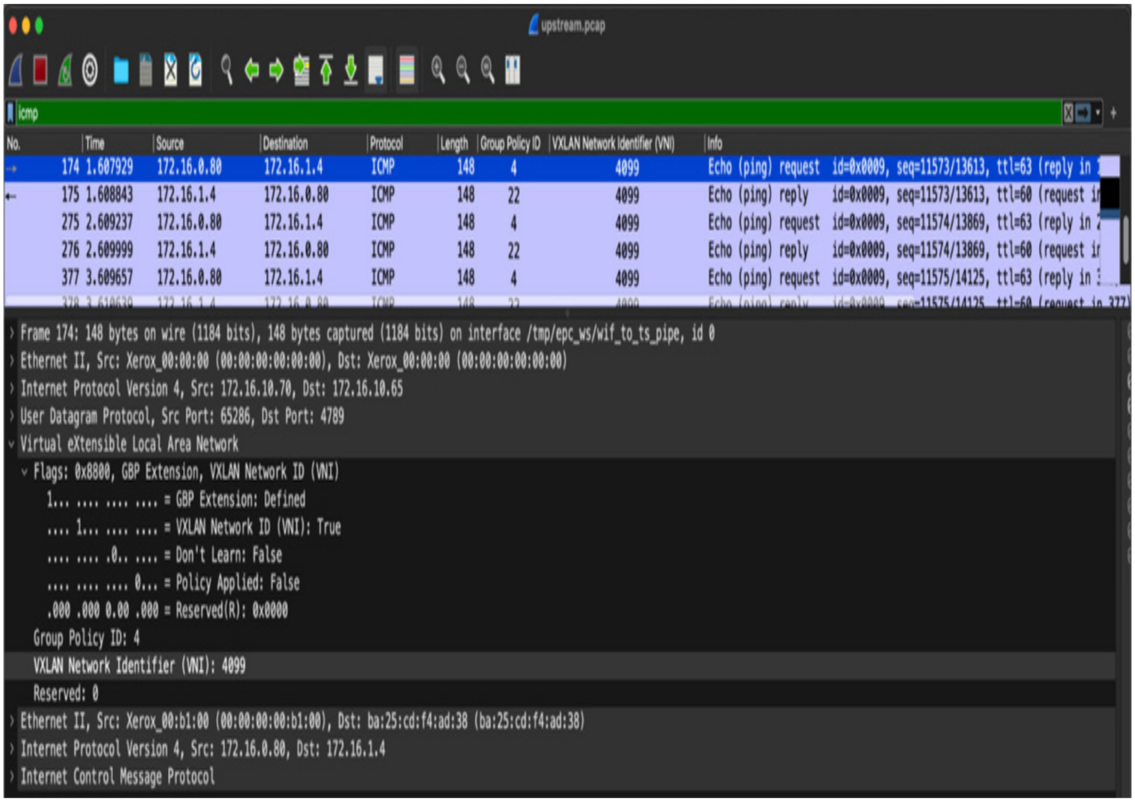


Figure 13-6 *Popular Layer 3 Constructs*

The traditional routed access design began with networks utilizing VLANs per access/edge switch stack, which would result in spanning tree avoidance. Newer constructs, which later came into production, added additional security layers into the architectures natively through the default deployment of VXLAN, allowing them to carry group-based policy and virtual network information inline on the data plane (see Figure 13-7).



```
No.     Time       Source        Destination   Protocol  Length  Group Policy ID  VXLAN Network Identifier (VNI)  Info
→   174 1.607929   172.16.0.80   172.16.1.4    ICMP      148     4                4099                            Echo (ping) request  id=0x0009, seq=11573/13613, ttl=63 (reply in 1
←   175 1.608843   172.16.1.4    172.16.0.80   ICMP      148     22               4099                            Echo (ping) reply    id=0x0009, seq=11573/13613, ttl=60 (request in
    275 2.609237   172.16.0.80   172.16.1.4    ICMP      148     4                4099                            Echo (ping) request  id=0x0009, seq=11574/13869, ttl=63 (reply in 2
    276 2.609999   172.16.1.4    172.16.0.80   ICMP      148     22               4099                            Echo (ping) reply    id=0x0009, seq=11574/13869, ttl=60 (request in
    377 3.609657   172.16.0.80   172.16.1.4    ICMP      148     4                4099                            Echo (ping) request  id=0x0009, seq=11575/14125, ttl=63 (reply in 3
    378 3.610630   172.16.1.4    172.16.0.80   ICMP      148     22               4099                            Echo (ping) reply    id=0x0009, seq=11575/14125, ttl=60 (request in 377)
```

```
> Frame 174: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits) on interface /tmp/epc_ws/wif_to_ts_pipe, id 0
> Ethernet II, Src: Xerox_00:00:00 (00:00:00:00:00:00), Dst: Xerox_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 172.16.10.70, Dst: 172.16.10.65
> User Datagram Protocol, Src Port: 65286, Dst Port: 4789
v Virtual eXtensible Local Area Network
  v Flags: 0x8800, GBP Extension, VXLAN Network ID (VNI)
      1... .... .... .... = GBP Extension: Defined
      .... 1... .... .... = VXLAN Network ID (VNI): True
      .... .... .0.. .... = Don't Learn: False
      .... .... .... 0... = Policy Applied: False
      .000 .000 0.00 .000 = Reserved(R): 0x0000
    Group Policy ID: 4
    VXLAN Network Identifier (VNI): 4099
    Reserved: 0
> Ethernet II, Src: Xerox_00:b1:00 (00:00:00:00:b1:00), Dst: ba:25:cd:f4:ad:38 (ba:25:cd:f4:ad:38)
> Internet Protocol Version 4, Src: 172.16.0.80, Dst: 172.16.1.4
> Internet Control Message Protocol
```

Figure 13-7 *VXLAN Packet Example*

Security within Layer 3 and fabric architectures falls back on common principles of limiting access to users and systems that require it. In most modern fabric infrastructures, exposure of the global routing table to client devices is not recommended (or even permitted), ensuring that clients are separated via macrosegmentation (VRF) into their own independent domain.

Optimizing Wireless Networks

Wireless architectures have become the new standard for many types of office environments, from enterprise environments to even some defense and military sectors. Ubiquitous access with Wi-Fi has changed over the years from a nice-to-have to a clear expectation for any deployment. To achieve high resiliency and uptime, redundant AP placement plans are often designed to provide primary and secondary levels of coverage. Such network designs (often referred to as a *chess*, *checkerboard*, or *salt-and-*

pepper design, as shown in Figure 13-8) are commonplace within most corporate organizations.

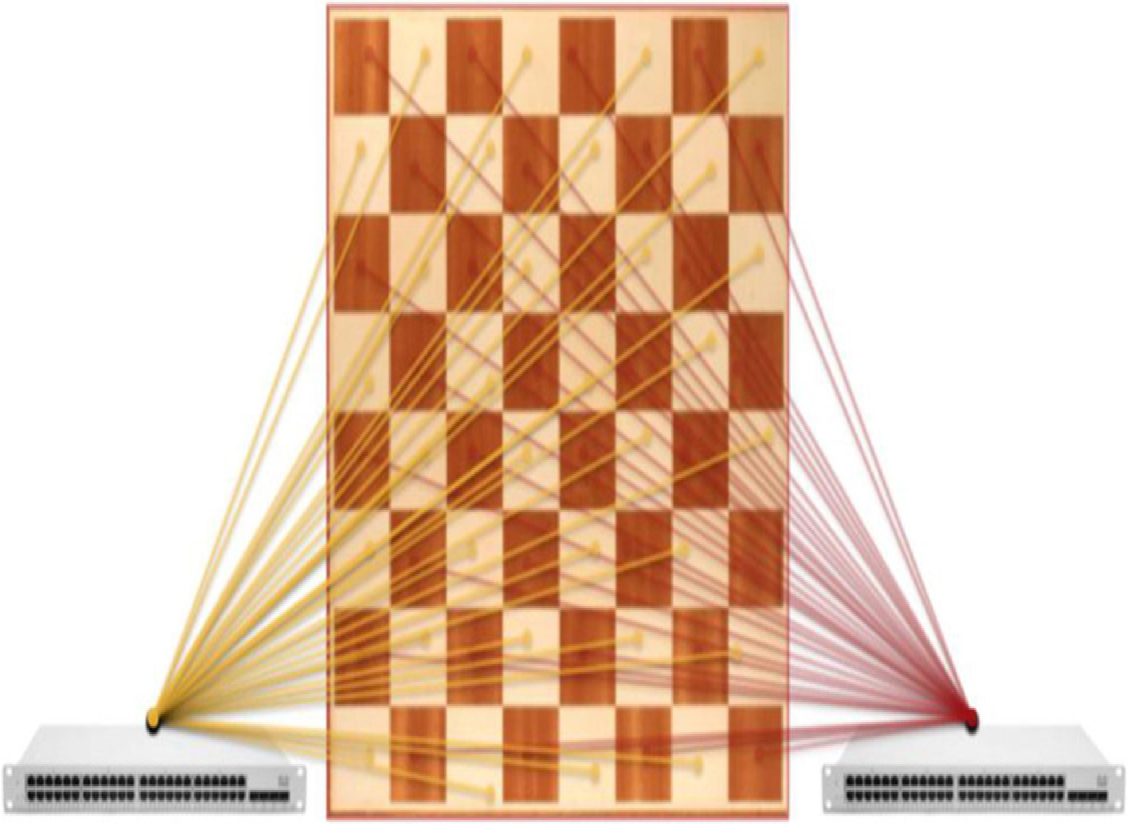
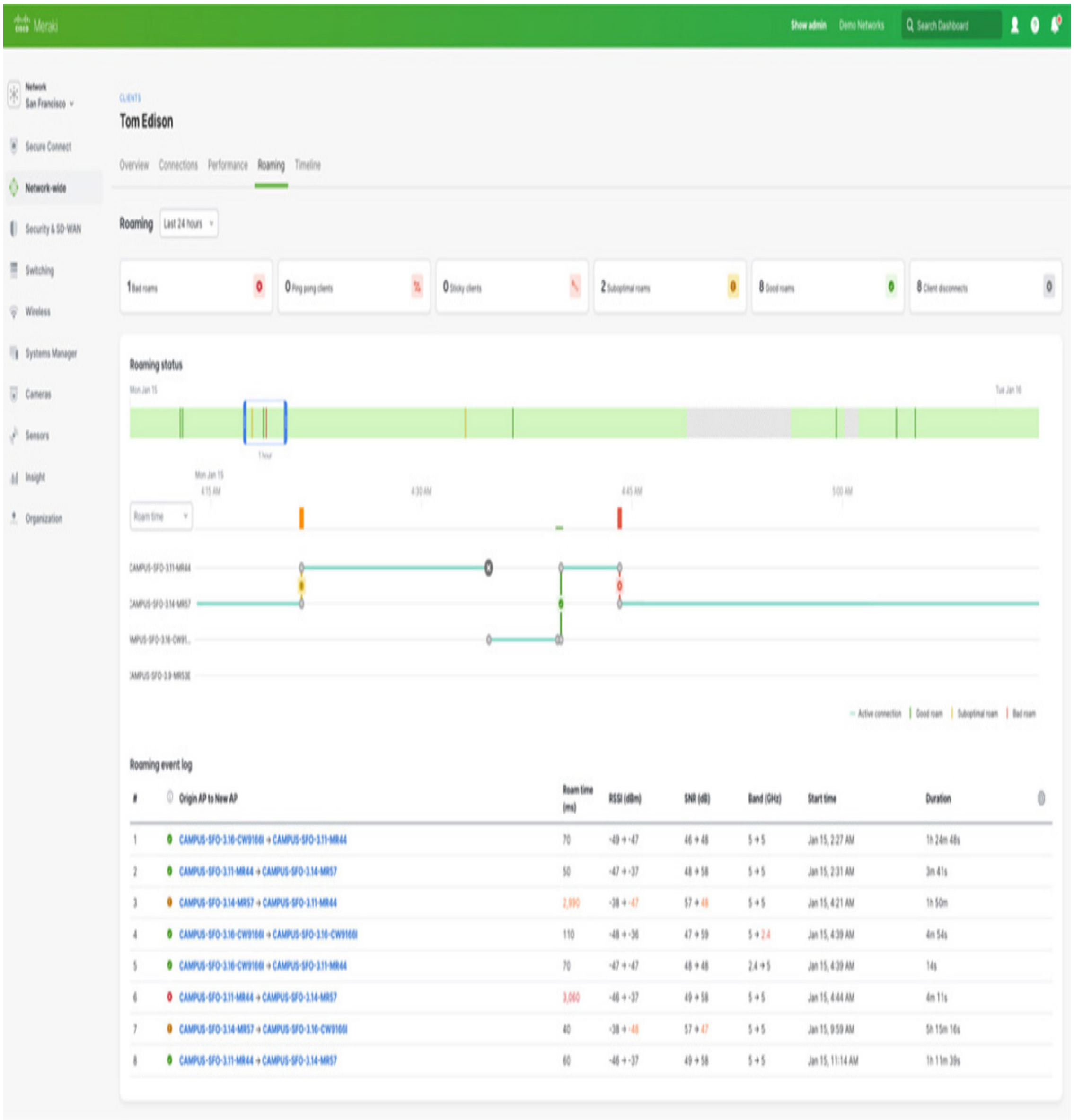


Figure 13-8 *Salt-and-Pepper AP Placement Design*

Because it is a key access medium, recommendations for Wi-Fi deployments have pivoted toward high security over the air, with client devices that are capable of leveraging WPA2 and, in newer environments, WPA3 Enterprise authentication methods in conjunction with fast roaming algorithms. Depending on the application needed and the speed at which roaming needs to take place, organizations should consider exactly which wireless authentication methods they should select. While some roaming methods may serve an office environment well, warehouses that are staffed by machines or autonomous vehicles that operate at high speed may exceed standard limits for certain protocols. Depending on the enterprise management solution chosen (Catalyst Center or Meraki), roaming data can be viewed from the dashboard, providing a perspective of how the client device is traversing the network.

This specific sort of data is important for an operator to evaluate and check on a regular basis, particularly after driver upgrades within the operation environments. Unhealthy systems or poor RF installations often result in excessive roaming events, which, if left unanalyzed, can result in a poor user experience. Figure 13-9 provides an overview of the Meraki dashboard's wireless roaming activity, vastly simplifying troubleshooting scenarios with roaming. This figure shows a time histogram matched to access points, showing where a specific client is moving and the duration taken when performing roaming events.



CLIENTS

Tom Edison

Roaming Last 24 hours

1 Bad roams | 0 Ping pong clients | 0 Sticky clients | 2 Suboptimal roams | 8 Good roams | 8 Client disconnects

Roaming event log

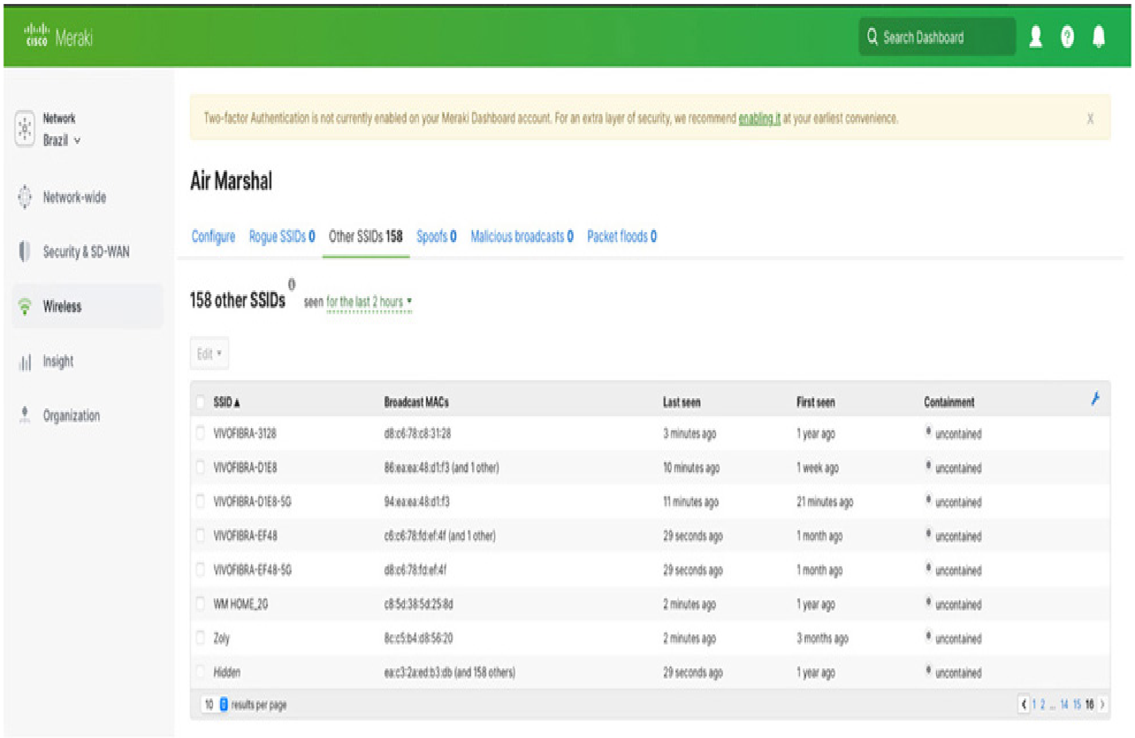
#	Origin AP to New AP	Roam time (ms)	RSSI (dBm)	SNR (dB)	Band (GHz)	Start time	Duration
1	● CAMPUS-SFO-3.16-CW9166I → CAMPUS-SFO-3.11-MR44	70	-49 → -47	46 → 48	5 → 5	Jan 15, 2:27 AM	1h 24m 48s
2	● CAMPUS-SFO-3.11-MR44 → CAMPUS-SFO-3.14-MR57	50	-47 → -37	48 → 58	5 → 5	Jan 15, 2:31 AM	3m 41s
3	● CAMPUS-SFO-3.14-MR57 → CAMPUS-SFO-3.11-MR44	2,990	-38 → -47	57 → 48	5 → 5	Jan 15, 4:21 AM	1h 50m
4	● CAMPUS-SFO-3.16-CW9166I → CAMPUS-SFO-3.16-CW9166I	110	-48 → -36	47 → 59	5 → 2.4	Jan 15, 4:39 AM	4m 54s
5	● CAMPUS-SFO-3.16-CW9166I → CAMPUS-SFO-3.11-MR44	70	-47 → -47	48 → 48	2.4 → 5	Jan 15, 4:39 AM	14s
6	● CAMPUS-SFO-3.11-MR44 → CAMPUS-SFO-3.14-MR57	3,060	-46 → -37	49 → 58	5 → 5	Jan 15, 4:44 AM	4m 11s
7	● CAMPUS-SFO-3.14-MR57 → CAMPUS-SFO-3.16-CW9166I	40	-38 → -48	57 → 47	5 → 5	Jan 15, 9:59 AM	5h 15m 16s
8	● CAMPUS-SFO-3.11-MR44 → CAMPUS-SFO-3.14-MR57	60	-46 → -37	49 → 58	5 → 5	Jan 15, 11:14 AM	1h 11m 39s

Figure 13-9 *Client Roaming Behavior*

Factors that may contribute to increased roaming in a Wi-Fi network environment are typically the result of client driver decisions, based on the network state assessing the radio signal strength and signal-to-noise ratio as a decision criterion toward roaming. Given that the Wi-Fi 802.11 standard utilizes the unlicensed spectrum in the 2.4 GHz, 5 GHz, and 6 GHz frequency ranges, interference from external sources is not uncommon, and client devices and networks need to be dynamic enough to adapt accordingly.

Common sources of interference are neighboring networks, which may be utilizing the same Wi-Fi channels. In busy cities and in buildings that host many different companies, it is not uncommon to have other devices that belong to third parties interfering with the network quality, lowering the available radio frequency throughput.

In both the Catalyst Center and Meraki dashboards, there are views available to observe the service set identifiers (SSIDs) that are being propagated in close proximity to the managed wireless network. In Figure 13-10, the Air Marshal in the Meraki dashboard provides an overview of wireless SSIDs and BSSIDs that are being heard by managed access points. The data shown in the dashboard can support a network operator in identifying the potential source of interference that may be affecting network performance.



Two-factor Authentication is not currently enabled on your Meraki Dashboard account. For an extra layer of security, we recommend enabling it at your earliest convenience.

Air Marshal

Configure | Rogue SSIDs 0 | Other SSIDs 158 | Spoofs 0 | Malicious broadcasts 0 | Packet floods 0

158 other SSIDs seen for the last 2 hours

SSID ▲	Broadcast MACs	Last seen	First seen	Containment
VIVOFIBRA-3128	d8:c6:78:c8:31:28	3 minutes ago	1 year ago	uncontained
VIVOFIBRA-D1E8	86:ea:ea:48:d1:f3 (and 1 other)	10 minutes ago	1 week ago	uncontained
VIVOFIBRA-D1E8-5G	94:ea:ea:48:d1:f3	11 minutes ago	21 minutes ago	uncontained
VIVOFIBRA-EF48	c6:c6:78:fd:ef:4f (and 1 other)	29 seconds ago	1 month ago	uncontained
VIVOFIBRA-EF48-5G	d8:c6:78:fd:ef:4f	29 seconds ago	1 month ago	uncontained
WM HOME_2G	c8:5d:38:5d:25:8d	2 minutes ago	1 year ago	uncontained
Zoly	8c:c5:b4:d8:56:20	2 minutes ago	3 months ago	uncontained
Hidden	ea:c3:2a:ed:b3:db (and 158 others)	29 seconds ago	1 year ago	uncontained

Figure 13-10 *Air Marshal Overview in Meraki*

While having external SSIDs working on the same channels can be annoying, it is not necessarily a malicious action. If, however, the corporate SSID that an organization is using is spoofed (copied), then this can represent a bigger problem for the organization. Often such attempts are performed by malicious actors who aim to harvest network data from a user or lead users to interface with portals that require them to enter secure credentials. The Cisco Meraki and Catalyst Center solutions can provide alarm options to identify such scenarios and support the network operator in tracking down and localizing rogue access points for removal.

A further option to remove malicious Wi-Fi APs and affected clients is *containment*, which is a mechanism that can be used to reduce the impact of an offending rogue access point that may be maliciously announcing the SSID. When configured to be active, containment will send 802.11 deauthentication frames with the spoofed source address of the rogue access point, prompting clients who may have accidently associated with the rogue network to disconnect. Figure 13-11 illustrates how containment takes place. While containment may not be able to fully remove a rogue device that is spoofing the known networks, it does mitigate a level of impact that such a device may cause.

Important!!

Legal considerations need to be taken into account with containment. A careful evaluation of whether containment is permitted in your country and state should be performed prior to activating this function in your network.

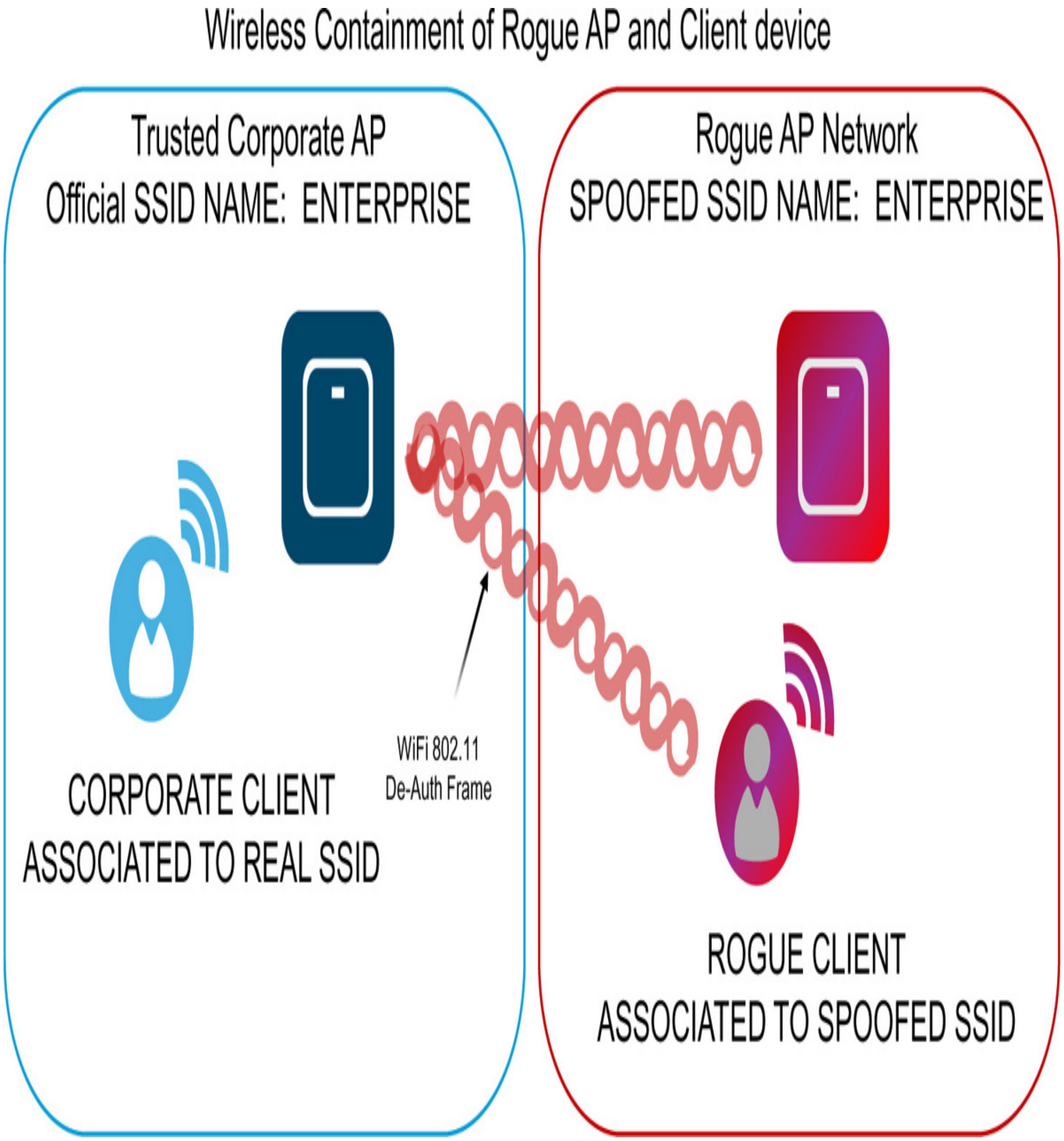

Wireless Containment of Rogue AP and Client device
Trusted Corporate AP
Official SSID NAME: ENTERPRISE
Rogue AP Network
SPOOFED SSID NAME: ENTERPRISE
WiFi 802.11
De-Auth Frame
CORPORATE CLIENT
ASSOCIATED TO REAL SSID
ROGUE CLIENT
ASSOCIATED TO SPOOFED SSID

Figure 13-11 *Wireless Containment of Rogue Devices*

Deployments of Wi-Fi using either the Cisco Catalyst Center or Meraki Cloud–based solution can provide an array of means to understand and verify the health of the RF environment, including monitoring spread spectrogram outputs. Figure 13-12 shows the RF utilization over time for the respective frequency bands. This view is useful in identifying whether the network was intentionally or unintentionally jammed via external interference. The spread spectrogram images that are shown at the top of the figure provide an active view of RF utilization across a specific radio frequency range; the lower portion of the image shows the spectrum use over time. This information can be helpful for an operator in identifying whether the radio frequency spectrum is persistently utilized or if sporadic events are occurring that may result in heavier utilization.

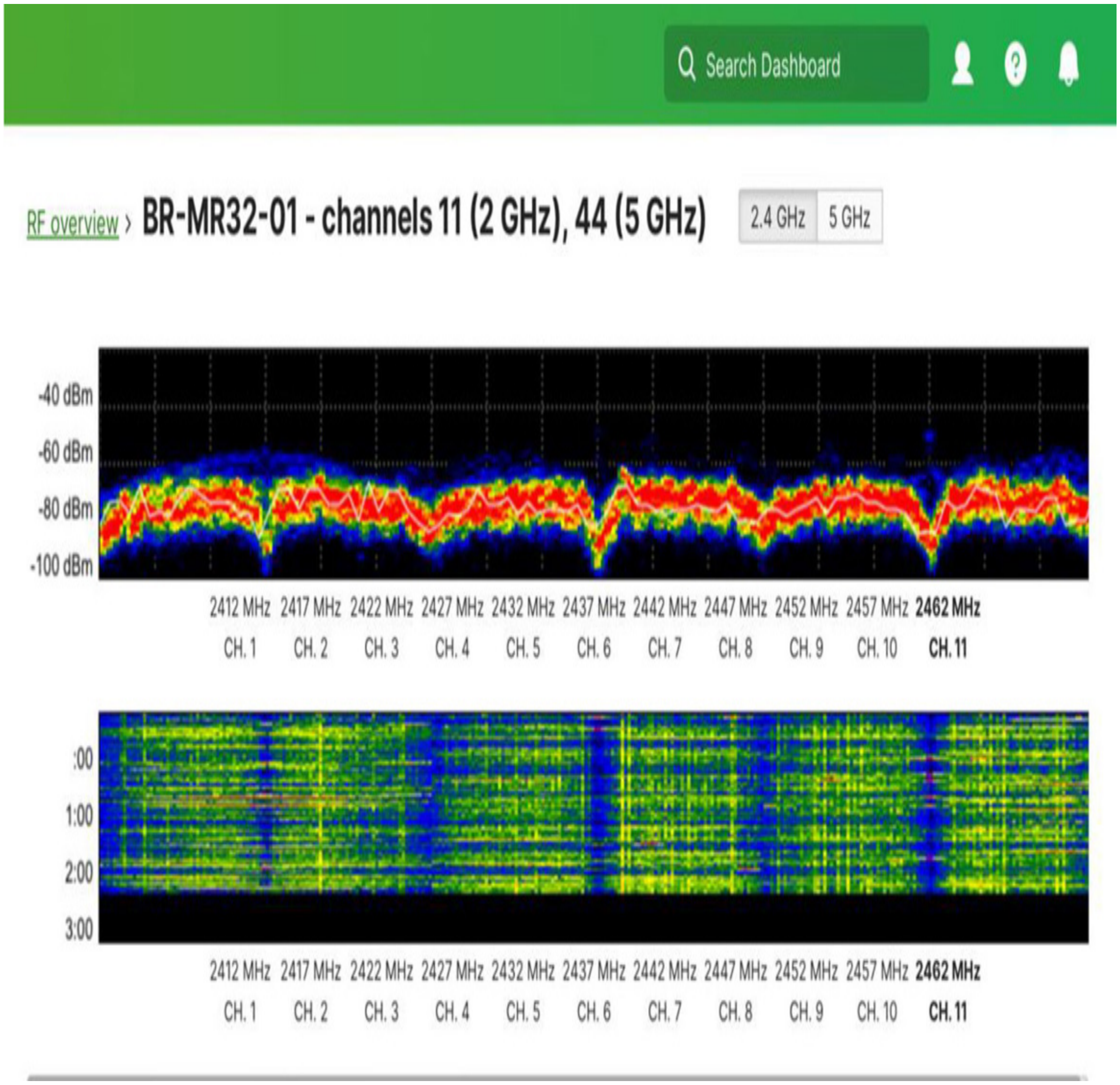
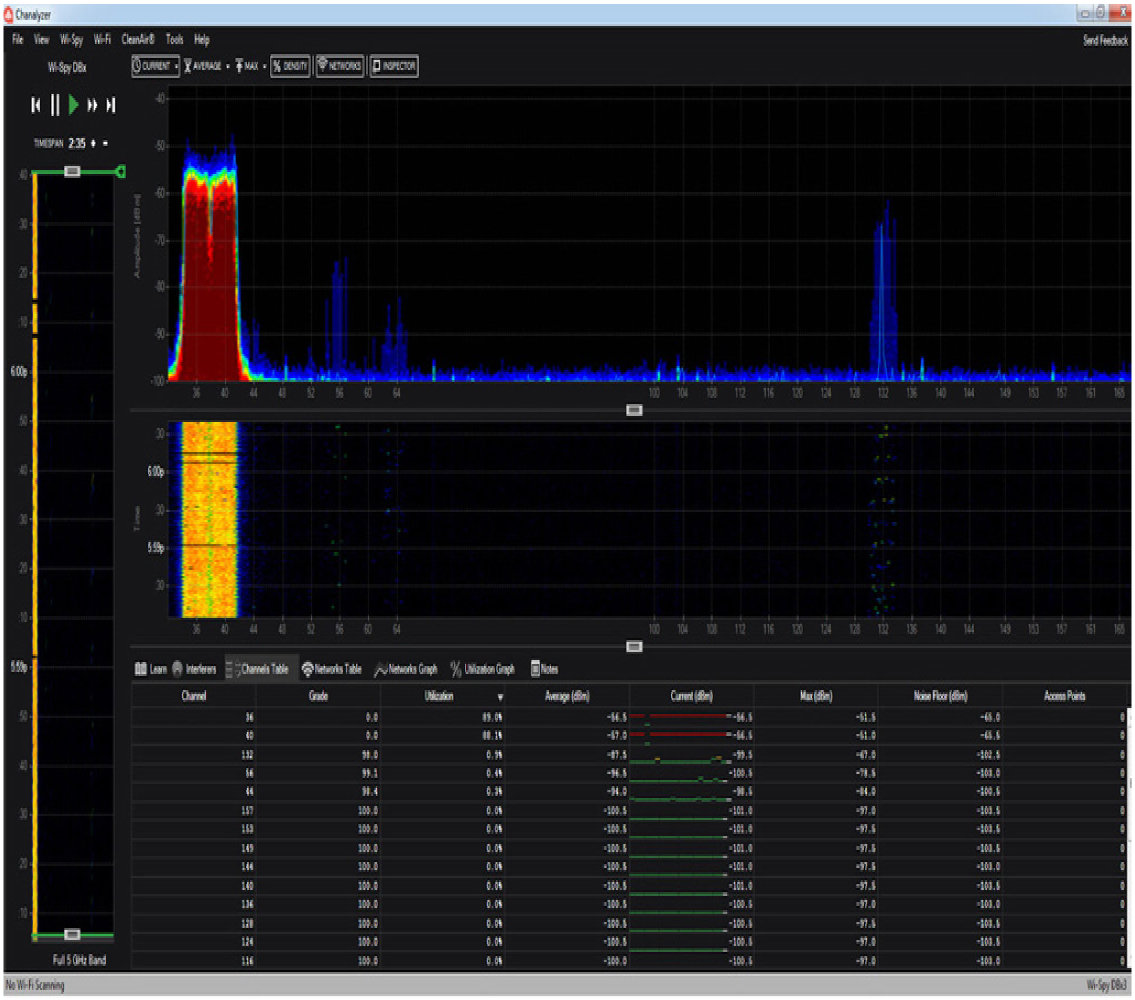

Search Dashboard
RF overview › BR-MR32-01 - channels 11 (2 GHz), 44 (5 GHz)
2.4 GHz
5 GHz
-40 dBm
-60 dBm
-80 dBm
-100 dBm
2412 MHz 2417 MHz 2422 MHz 2427 MHz 2432 MHz 2437 MHz 2442 MHz 2447 MHz 2452 MHz 2457 MHz 2462 MHz
CH. 1 CH. 2 CH. 3 CH. 4 CH. 5 CH. 6 CH. 7 CH. 8 CH. 9 CH. 10 CH. 11
:00
1:00
2:00
3:00
2412 MHz 2417 MHz 2422 MHz 2427 MHz 2432 MHz 2437 MHz 2442 MHz 2447 MHz 2452 MHz 2457 MHz 2462 MHz
CH. 1 CH. 2 CH. 3 CH. 4 CH. 5 CH. 6 CH. 7 CH. 8 CH. 9 CH. 10 CH. 11

Figure 13-12 *Meraki Dashboard Spread Spectrogram*

Figure 13-13 shows an example of jammed radio channels (channels 36 and 40) in contrast to a healthy and sparsely utilized RF environment. The high utilization that is shown in the top part of the window from the Channelizer spectrum measurement application indicates that the radio frequency is heavily consumed. As the levels show, it is almost impossible for normal data communications to take place. Much like the bottom part of the image from the Meraki Dashboard in Figure 13-12, this figure shows a view over time, showing that this heavy utilization has been persistent for radio channels 36 and 40 but has been very low in terms of consumption in other channels in the 5 GHz Wi-Fi radio spectrum range.



Channel	Grade	Utilization	Average (dBm)	Current (dBm)	Max (dBm)	Noise Floor (dBm)	Access Points
36	0.0	89.0%	-56.5	-56.5	-51.5	-65.0	0
40	0.0	88.1%	-57.0	-56.5	-51.0	-65.5	0
132	98.0	0.9%	-97.5	-99.5	-67.0	-102.5	0
56	99.1	0.4%	-96.5	-100.5	-78.5	-103.0	0
44	98.4	0.3%	-94.0	-98.5	-84.0	-100.5	0
157	100.0	0.0%	-100.5	-101.0	-97.0	-103.5	0
153	100.0	0.0%	-100.5	-101.0	-97.5	-103.5	0
149	100.0	0.0%	-100.5	-101.0	-97.5	-103.5	0
144	100.0	0.0%	-100.5	-101.0	-97.5	-103.0	0
140	100.0	0.0%	-100.5	-101.0	-97.5	-103.5	0
136	100.0	0.0%	-100.5	-100.5	-97.0	-103.0	0
128	100.0	0.0%	-100.5	-100.5	-97.5	-103.5	0
124	100.0	0.0%	-100.5	-100.5	-97.0	-103.5	0
116	100.0	0.0%	-100.0	-100.5	-97.0	-103.0	0

Figure 13-13 *Jammed RF Channel*

Understanding the ins and outs of radio frequency utilization doesn't happen overnight, but given the tools available in the Catalyst Center, Meraki Dashboard, and external analyzer tools such as Metageek

Channelizer Pro, Ekahau Site Survey, and tools from Hamina, the opportunity for network operators to view, understand, and familiarize themselves with spectrum analysis becomes a lot easier.

In regard to wireless networks, you should consider that the choice of mediums can range from short-range beacon technologies, such as ZigBee and BLE, to the various flavors of 802.11 Wi-Fi technologies that exist, to longer-range communication mediums, such as Private 5G and LoRaWAN. Each of these technologies utilizes radio frequencies to achieve a given use case. Some use cases and needs are considered more short range, whereas others provide broad and wide ranges of deployability.

One constant that needs to exist, regardless of the chosen RF medium, is the right level of security and control over the construct from an infrastructure and client perspective that is intended to be deployed. Wi-Fi solutions exist to match up with any sort of deployment scenario, ranging from on-premises setups that utilize physical wireless LAN controllers, to cloud-based offerings such as the Cisco Meraki portfolio.

When you look beyond deploying only classic Wi-Fi installations, various radio frequency communications mediums can co-exist, with communications traversing between client devices that may be deployed in either network medium at a given time. Often, decisions for secondary coverage domains—or micro coverage domains in the case of beacons—come down to the needs and requirements of the application landscape.

If high bandwidth rates are not an immediate requirement for the installation, the very large cell size that private 5G can provide is often very attractive for customers, particularly in the industrial domain, where large plants and facilities require blanket coverage over broad spaces.

Figure 13-14 breaks down where strengths and weaknesses can be observed with different RF technologies.

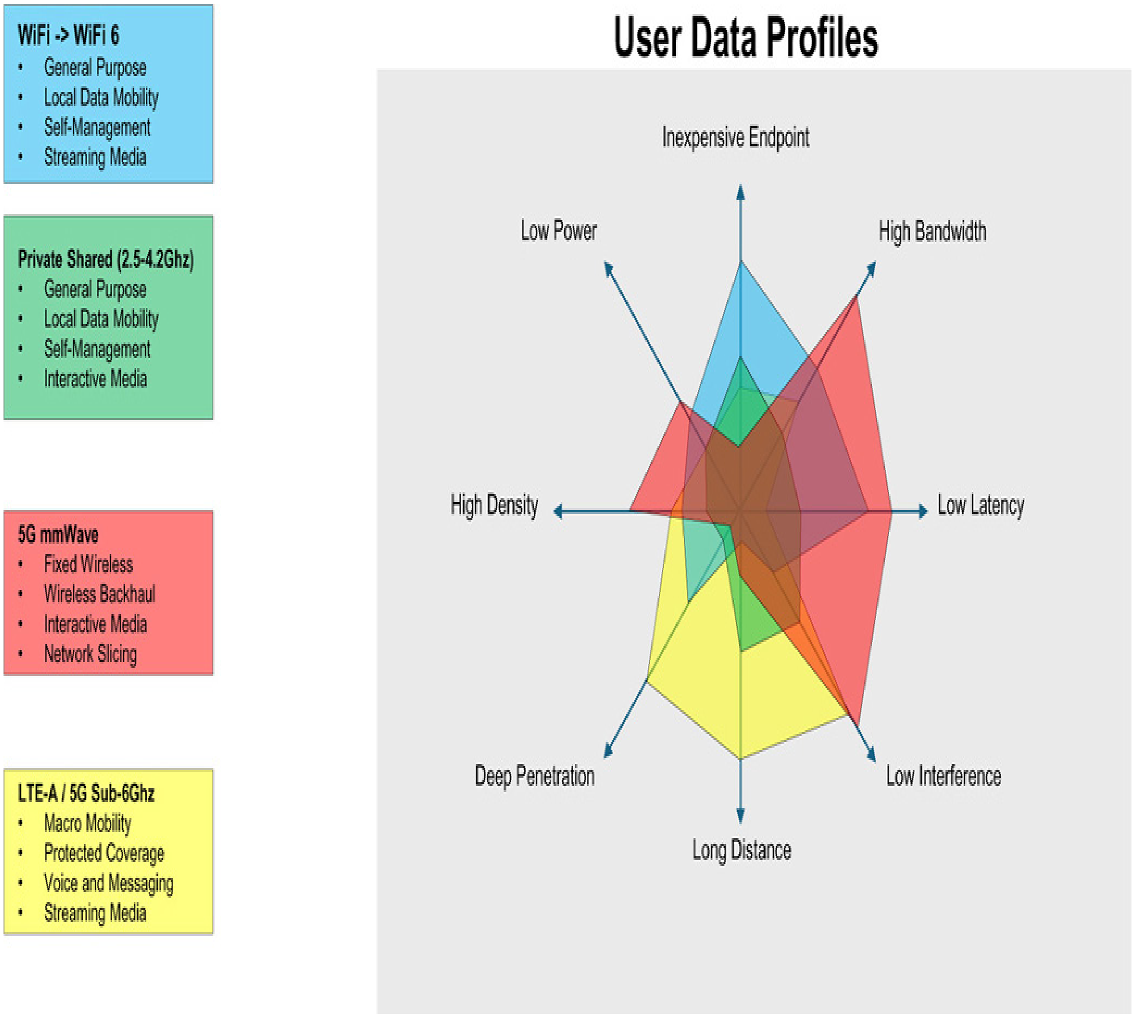


Figure 13-14 *RF Technology Strengths and Weaknesses*

In addition to the coverage aspects that private 5G may offer, from an authentication and authorization aspect, subscriber management is handled through different methods in private 5G than with Wi-Fi. Despite the variations in how mobile endpoints connect to the network, within the Cisco solution, there is the ability to execute secondary authorization, as shown in Figure 13-15, providing the user with common policy using Identity Services Engine (ISE).

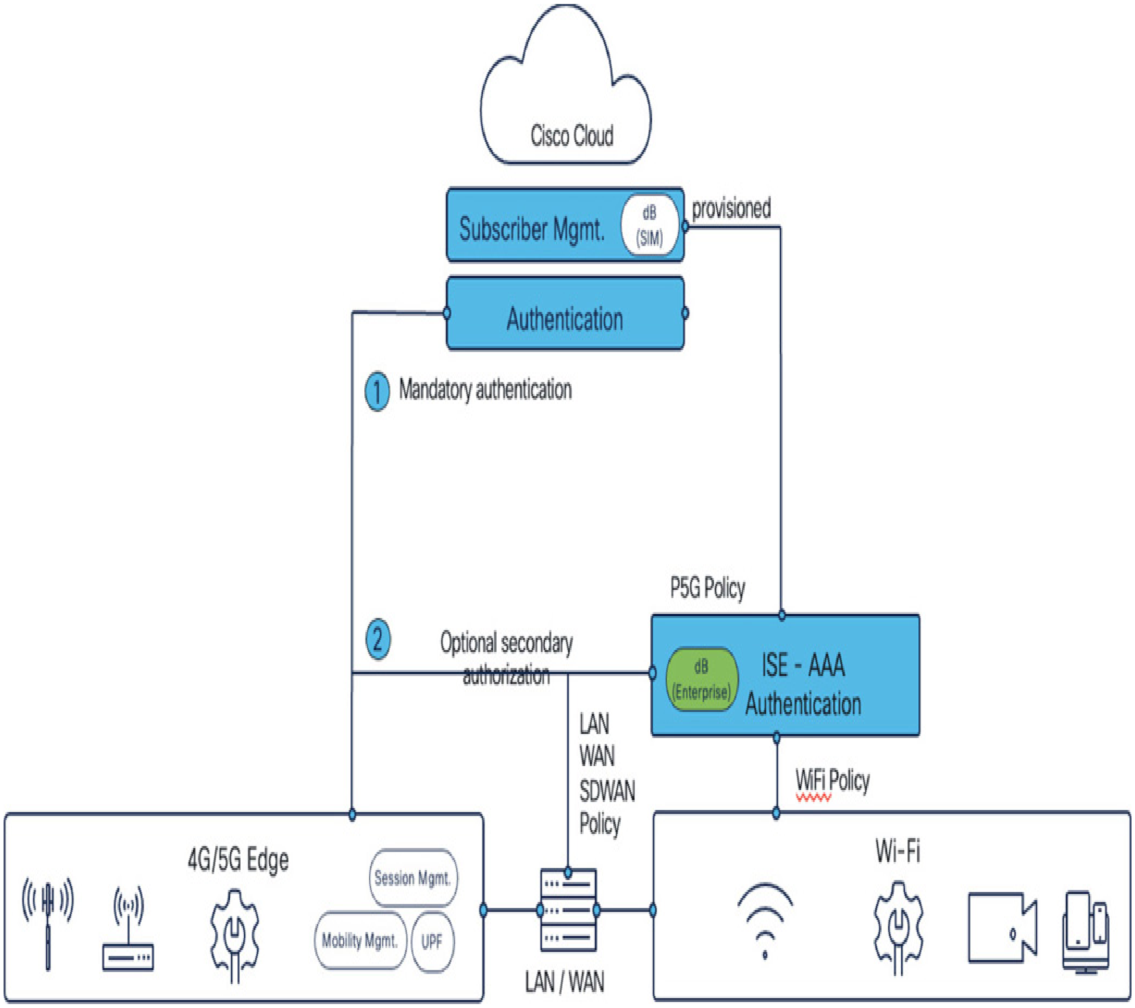


Figure 13-15 *Secondary Authorization for Private 5G*

The steps associated with secondary authorization are shown in more granular detail in Figure 13-16, which showcases the numeric sequence of steps that are performed to achieve secondary authorization for private 5G devices.

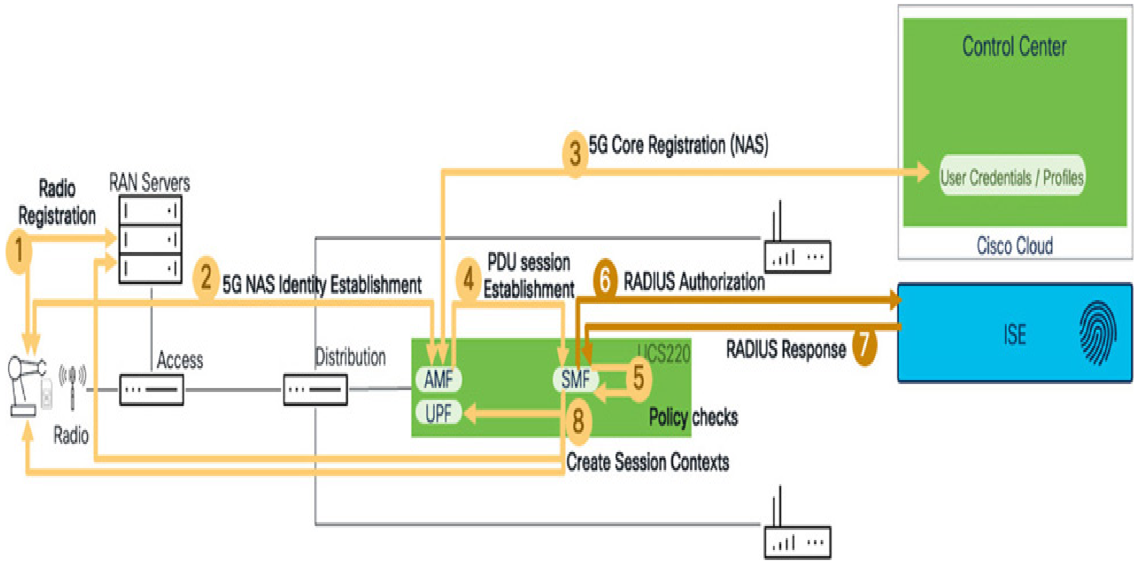


Figure 13-16 *Private 5G Secondary Authorization Sequence*

Using the described methods, an organization can apply complex policies and rulesets to network endpoints, including the deployment of microsegmentation through TrustSec tag allocations and enforcement.

When you are considering a broader network architecture, the use of wired, Wi-Fi, private 5G, and VPN connectivity, leveraging common classification and enforcement mechanisms can bring a broad range of benefits in terms of security enforcement and visibility of communications.

The deployment options that are most evaluated when deploying a Wi-Fi network generally land in two main categories; each category has advantages and disadvantages that need to be considered.

Central Tunneling of Traffic (Over the Top)

These constructs have been a common tenet of Wi-Fi deployments for many years, often having the advantages of providing a central breakout point and depending on the hardware and software in use, a potentially more rapid capability for roaming and failover. The use of tunneling is often also considered when identifying a secure location to offload corporate wireless clients or insecure guest devices.

When wireless controllers or concentrators in Meraki are placed at a distance from the site, challenges with this construct are, of course, the carriage of the client traffic that would need to be extended to the location where the controller resides. This sort of architecture, while still used by many customers today, is less common than it has been historically, due to the desire to have a comparable experience between wired and wireless in terms of IP subnet termination.

The DHCP infrastructure typically exists northbound from the wireless controller or concentrator architecture or can be reached through requests proxied by the controller.

Local Breakout

Initially deployed in scenarios such as FlexConnect or bridge mode in Meraki deployments, the architecture where client devices break out to the access switch topology is becoming more popular, allowing client devices to share the same IP pools or ranges as other network mediums such as wired.

With the introduction of modern fabric architectures such as the LISP-based SDA architecture and the BGP-based EVPN fabric architecture, the reduction of broadcast traffic within switching blocks has also fostered the increased adoption of this method in many customer environments. IP address allocation in local breakout setups can take place through the use of functions such as IP Helper or similar relay technologies or, in Meraki devices, the use of NAT mode.

Anchoring Concepts (Catalyst/Meraki)

The concept of *anchoring* is relatively straightforward; it maps quite closely to the physical world, in that the right place for an anchor to exist when out at sea is the ocean floor, far away from the vessel. This same principle applies to insecure clients or devices from both wired and wireless domains. The objective is to keep the insecure domain at a distance from the secure domain. There are numerous ways in which this can be achieved. The decision of how and where to anchor such traffic is often heavily linked to operational requirements and needs from the architecture and relevant corporate security departments within the respective organization that the deployment is planned for.

Classic anchoring, which is commonly used today in Catalyst Wi-Fi deployments, is established through two sets of tunnels (see Figure 13-17). The initial client traffic traverses a CAPWAP tunnel, from AP to foreign wireless controllers, at which point the connection is reencapsulated into a secondary Ethernet over IP (EoIP) tunnel toward the anchor controller, which typically resides in an insecure network segment such as a demilitarized zone (DMZ).

Insecure devices, often guest clients, receive an IP address in the DMZ at the anchor controller. In most anchor scenarios that use this construct, traffic from insecure clients in the DMZ back into the secure network is forbidden for security reasons.

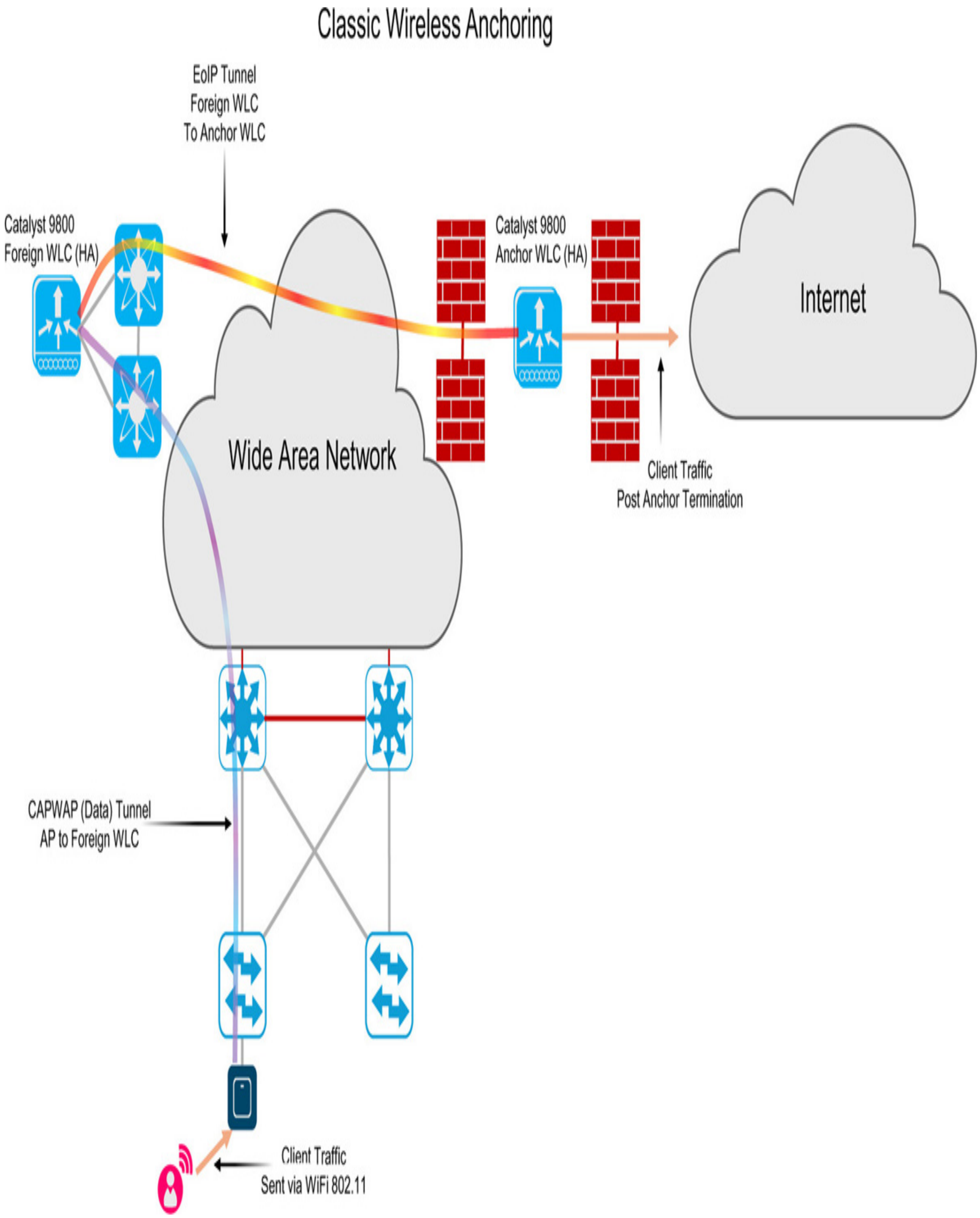


Figure 13-17 *Classic Anchoring (Catalyst)*

In SD-Access networks, the concept of a multi-site remote border (MSRB) exists: this construct bears a number of similarities to the previous classic anchoring scenario that was described. One key difference is that the traffic from the client is not reencapsulated at a wireless controller to be forwarded. Because this construct is specific to SD-Access networks, the traffic from the client at no point in time traverses a wireless LAN controller; rather, it uses the VXLAN, from its point of origin, to reach its local site exit point (an SDA border node), at which point it is reencapsulated into a second VXLAN tunnel and sent to the multi-site remote border.

This construct has a number of advantages over the classic deployment construct, in that the traffic is not hairpinned at the wireless LAN controller. However, due to the way that SD-Access works, by utilizing a LISP-based control plane, roaming events for traffic flows that may be anchored at a multi-site remote border that is geographically distant from the site itself may be slower than expected due to increased LISP control plane traffic, which is required to be in the site border node and the DMZ that hosts the MSRB. See Figure 13-18.

IP addressing constructs in a multi-site remote border configuration can be common between all sites when needed.

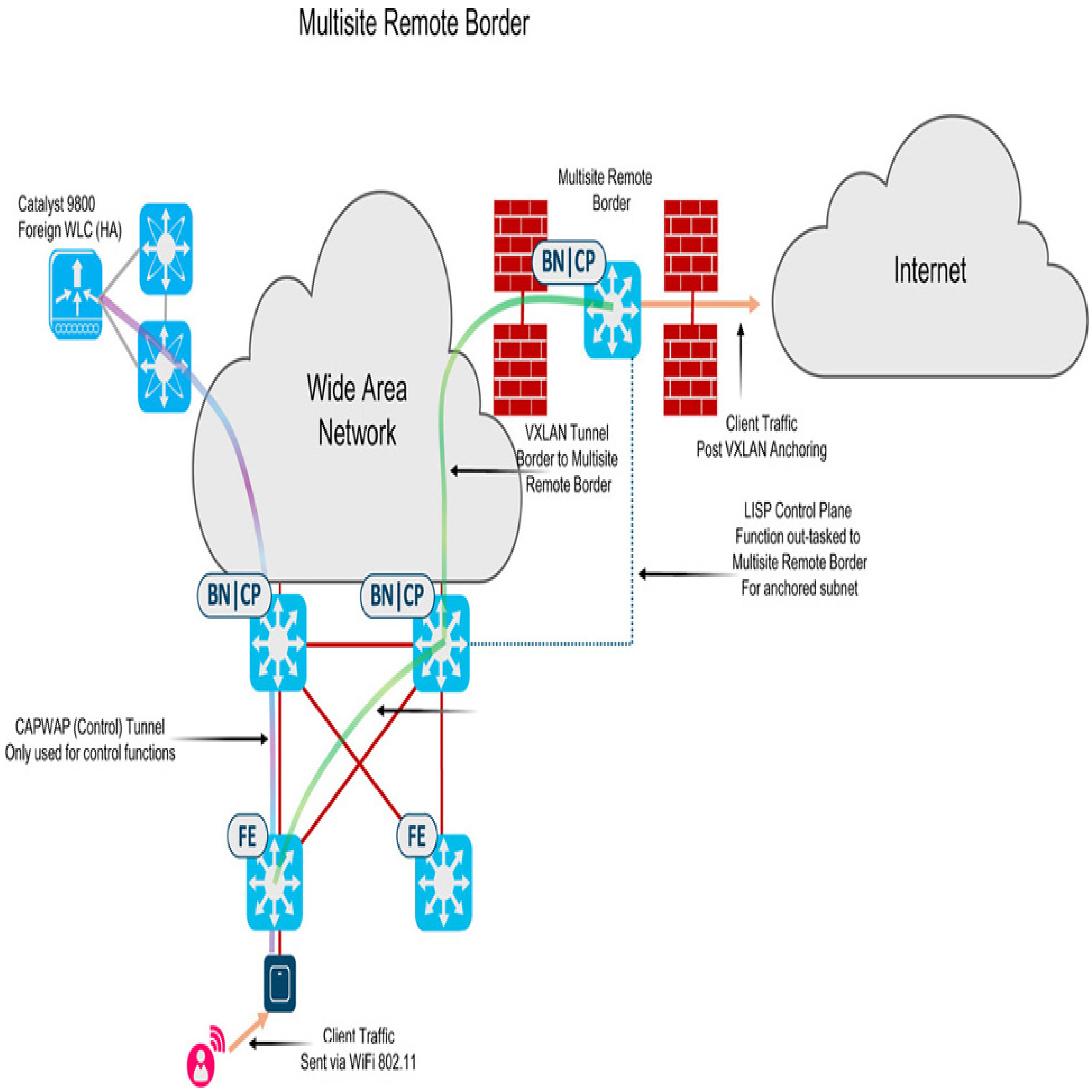


Figure 13-18 *Multi-Site Remote Border*

The anchor remote border method retains similarities to the other previously described constructs (see Figure 13-19); however, unlike the multi-site remote border, which requires heavy control plane communications for its clients, this construct makes use of the SDA transit deployment constructs, with insecure client devices using the SDA transit borders that have the **Connected to Internet** option configured in Catalyst Center to allocate them as exit points geographically.

This setup is much like a multi-site remote border because this is an SD-Access network construct; however, there is not an option to use the same

IP subnet across sites. Instead, a subnet (or subnets) per site needs to be used in this constellation.

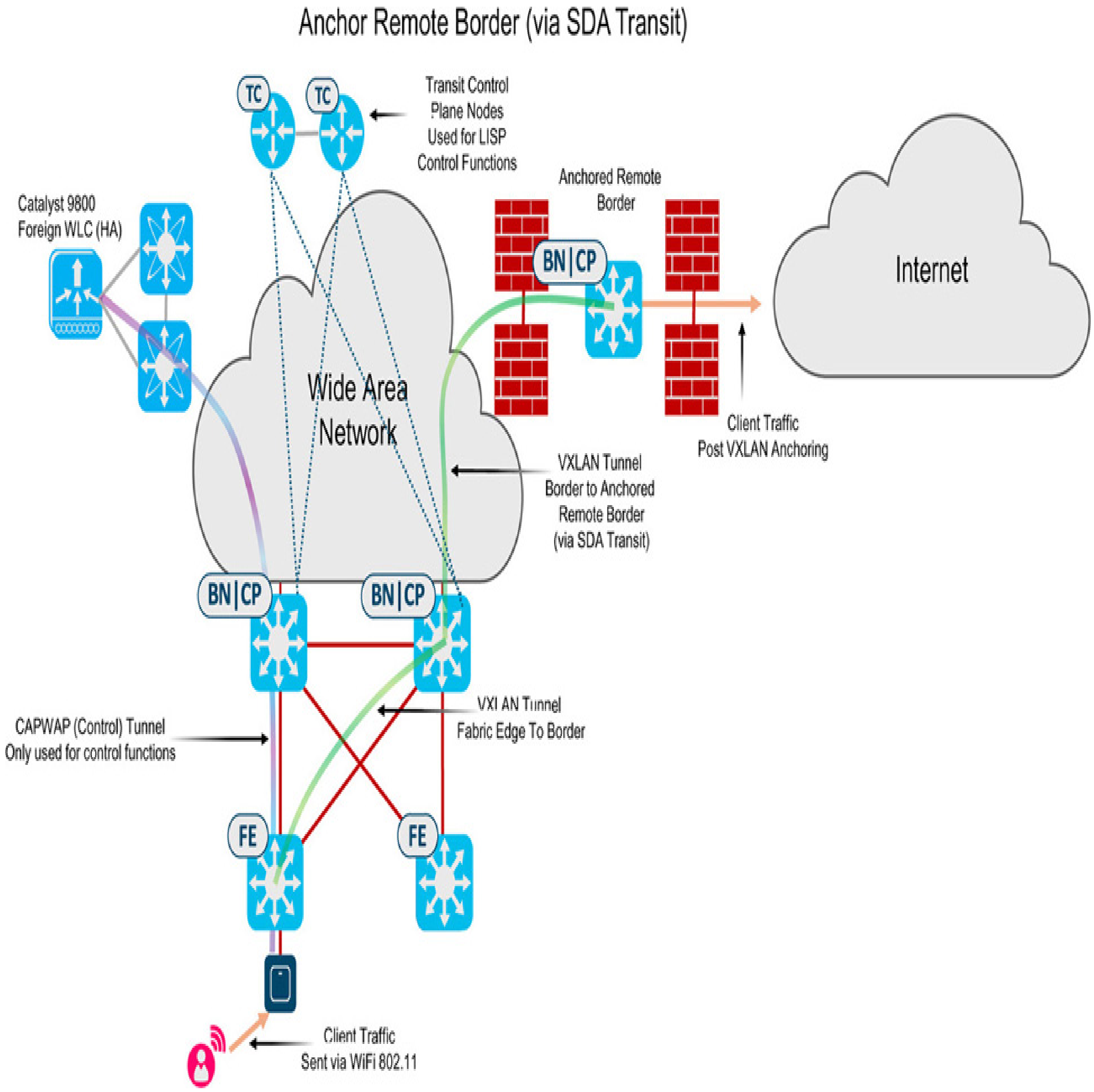

Anchor Remote Border (via SDA Transit)
TC
TC
Transit Control Plane Nodes Used for LISP Control Functions
Catalyst 9800 Foreign WLC (HA)
Anchored Remote Border
BN|CP
Internet
Wide Area Network
Client Traffic Post VXLAN Anchoring
VXLAN Tunnel Border to Anchored Remote Border (via SDA Transit)
BN|CP
BN|CP
VXLAN Tunnel Fabric Edge To Border
CAPWAP (Control) Tunnel Only used for control functions
FE
FE
Client Traffic Sent via WiFi 802.11

Figure 13-19 *Anchor Remote Border*

In Meraki deployments, further anchoring options are available to configure, which also provide a means of sending traffic to a central breakout point for secure separation from other SSIDs. The most common option available in the Meraki portfolio today is using an MX Concentrator device. Traffic that originates at the access point terminates at the concentrator, allowing for a central breakout point. Figure 13-20 showcases the flow associated with anchoring to a Meraki MX Concentrator.

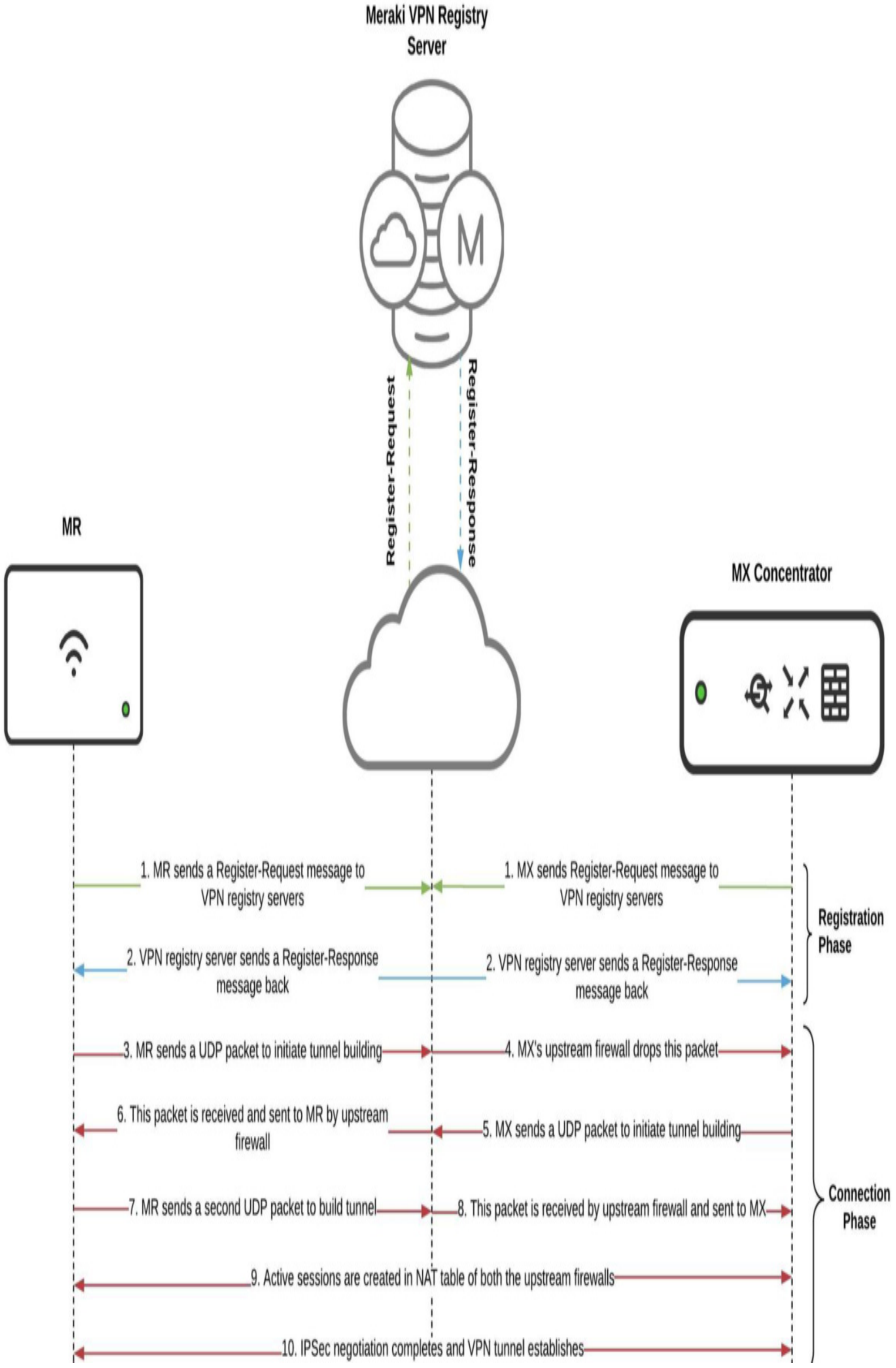

Meraki VPN Registry Server
Register-Request
Register-Response
MR
MX Concentrator
1. MR sends a Register-Request message to VPN registry servers
1. MX sends Register-Request message to VPN registry servers
2. VPN registry server sends a Register-Response message back
2. VPN registry server sends a Register-Response message back
Registration Phase
3. MR sends a UDP packet to initiate tunnel building
4. MX's upstream firewall drops this packet
6. This packet is received and sent to MR by upstream firewall
5. MX sends a UDP packet to initiate tunnel building
7. MR sends a second UDP packet to build tunnel
8. This packet is received by upstream firewall and sent to MX
9. Active sessions are created in NAT table of both the upstream firewalls
10. IPSec negotiation completes and VPN tunnel establishes
Connection Phase

Figure 13-20 *Meraki Concentrator Anchoring*

Monitoring TrustSec and Security Enforcement

When you’re deploying a TrustSec security group tag–based enforcement architecture (introduced in earlier chapters) across wired, Wi-Fi, or private 5G-based network mediums, it is important to have the right level of monitorability in place. Much like the deployment of a perimeter firewall, without the ability to verify and confirm that the communications are intact, deployment and operations can become cumbersome.

Day-to-day challenges that operators face, when dealing with security enforcement layers of architectures, include understanding if the submitter of a policy for a respective service has indeed requested the right communication matrix. Security rule requests tend to consist of an array of source-to-destination tuples, including source ports and destination ports that represent the foundation requirement for a service to operate.

Quite often a challenge that is observed, particularly in enterprise domains, is that the IoT or client/server-side applications may undergo an upgrade or change of software, which in turn represents an addition or change of network ports that are being used. Utilizing the audit trail on a security appliance can enable you to better understand permits and drops in this context.

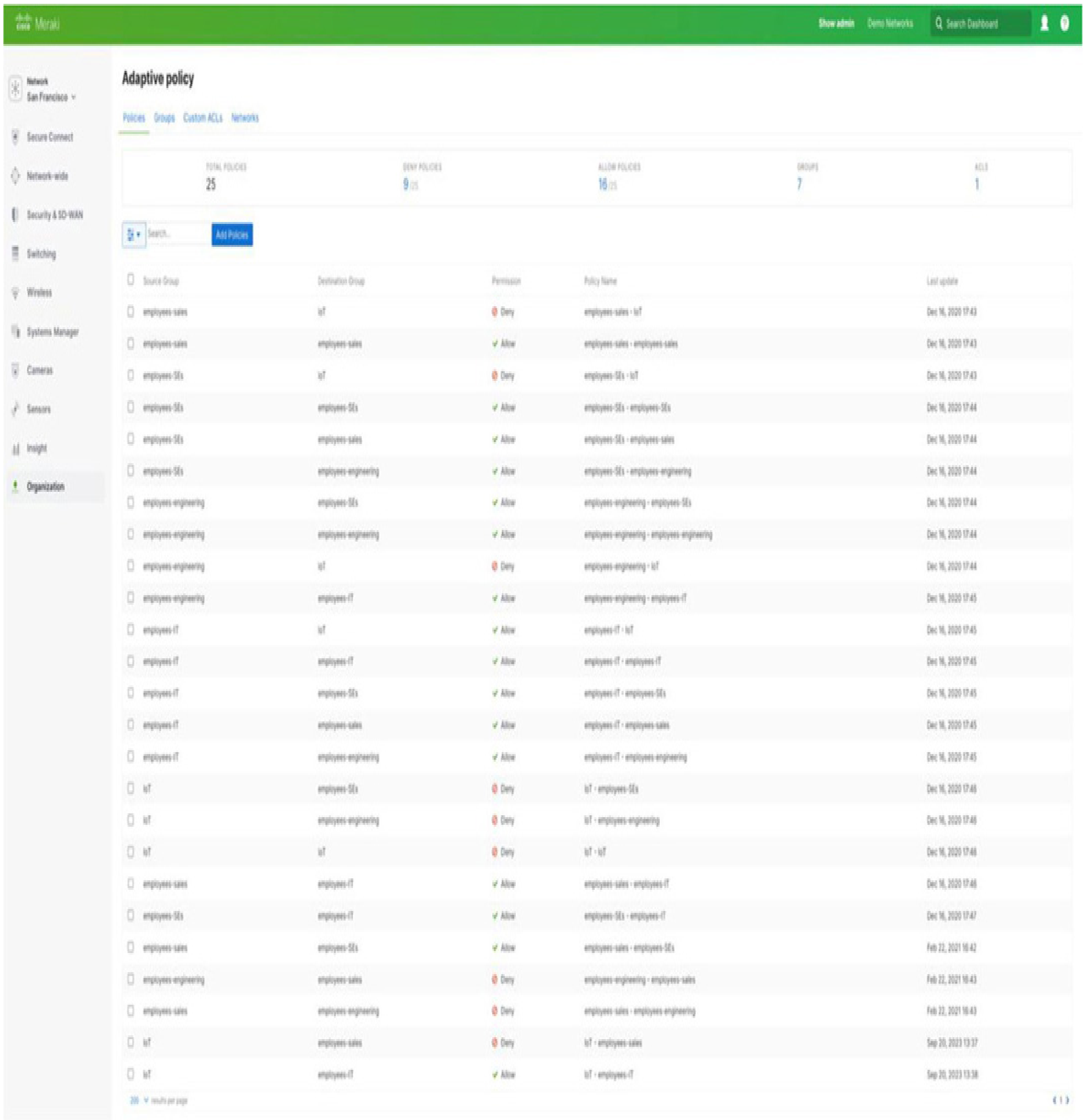
When looking into how the same approach may be achieved in the context of TrustSec, however, depending on the deployment that is in place, this data can be visualized and consumed in different ways to gain the right level of operational visibility.

Having the right policy in place for north-to-south and east-to-west communications within the network ensures that client and network resources are only communicating with applications and systems that should be explicitly permitted. Once that policy is active, verifying hits, misses, thresholds, and anomalies becomes part of the daily operating procedures for a security team.

As with any security policy, an active review and assessment of the requirements may be needed for approval of deployment within a secure network. Despite the best efforts of the policy requestors, misses and gaps in the rule base are often observed. These gaps may be due to vendor specifications differing from new software versions and the documentation no longer being up-to-date, or simply the result of incorrect submissions.

If you need to adequately monitor and react to issues with a new policy, or to evaluate the need for adjustments to be made, an up-to-date view of how the data is traversing the network is key. You can achieve this through a number of different methods.

The first step in assessment is to understand what the existing policy looks like. You can perform such steps in the **Adaptive Policy** tab in Meraki (see Figure 13-21), directly in ISE in the TrustSec Matrix, or in the **Group Based Access Control** section in Catalyst Center.



Adaptive policy

TOTAL POLICIES	DENY POLICIES	ALLOW POLICIES	GROUPS	ACLS
25	9/25	16/25	7	1

Source Group	Destination Group	Permission	Policy Name	Last update
employees-sales	IoT	Deny	employees-sales - IoT	Dec 16, 2020 17:43
employees-sales	employees-sales	✓ Allow	employees-sales - employees-sales	Dec 16, 2020 17:43
employees-SEs	IoT	Deny	employees-SEs - IoT	Dec 16, 2020 17:43
employees-SEs	employees-SEs	✓ Allow	employees-SEs - employees-SEs	Dec 16, 2020 17:44
employees-SEs	employees-sales	✓ Allow	employees-SEs - employees-sales	Dec 16, 2020 17:44
employees-SEs	employees-engineering	✓ Allow	employees-SEs - employees-engineering	Dec 16, 2020 17:44
employees-engineering	employees-SEs	✓ Allow	employees-engineering - employees-SEs	Dec 16, 2020 17:44
employees-engineering	employees-engineering	✓ Allow	employees-engineering - employees-engineering	Dec 16, 2020 17:44
employees-engineering	IoT	Deny	employees-engineering - IoT	Dec 16, 2020 17:44
employees-engineering	employees-IT	✓ Allow	employees-engineering - employees-IT	Dec 16, 2020 17:45
employees-IT	IoT	✓ Allow	employees-IT - IoT	Dec 16, 2020 17:45
employees-IT	employees-IT	✓ Allow	employees-IT - employees-IT	Dec 16, 2020 17:45
employees-IT	employees-SEs	✓ Allow	employees-IT - employees-SEs	Dec 16, 2020 17:45
employees-IT	employees-sales	✓ Allow	employees-IT - employees-sales	Dec 16, 2020 17:45
employees-IT	employees-engineering	✓ Allow	employees-IT - employees-engineering	Dec 16, 2020 17:45
IoT	employees-SEs	Deny	IoT - employees-SEs	Dec 16, 2020 17:46
IoT	employees-engineering	Deny	IoT - employees-engineering	Dec 16, 2020 17:46
IoT	IoT	Deny	IoT - IoT	Dec 16, 2020 17:46
employees-sales	employees-IT	✓ Allow	employees-sales - employees-IT	Dec 16, 2020 17:46
employees-SEs	employees-IT	✓ Allow	employees-SEs - employees-IT	Dec 16, 2020 17:47
employees-sales	employees-SEs	✓ Allow	employees-sales - employees-SEs	Feb 22, 2021 16:42
employees-engineering	employees-sales	Deny	employees-engineering - employees-sales	Feb 22, 2021 16:43
employees-sales	employees-engineering	Deny	employees-sales - employees-engineering	Feb 22, 2021 16:43
IoT	employees-sales	Deny	IoT - employees-sales	Sep 20, 2023 13:37
IoT	employees-IT	✓ Allow	IoT - employees-IT	Sep 20, 2023 13:38

Figure 13-21 *Adaptive Policy in Meraki Dashboard*

The first consideration that you need to make is where exactly the enforcement (permit/deny/partial permit) of the source-to-destination communication is taking place. Unlike static or dynamic access lists, enforcement takes place on the egress; this means that the traffic traverses from the source device to the destination, throughout the network, until the last hop of TrustSec enforcement to be dropped.

The examples provided in Figure 13-22 showcase some of the specifics that need to be observed when attempting to enforce TrustSec within a network architecture.

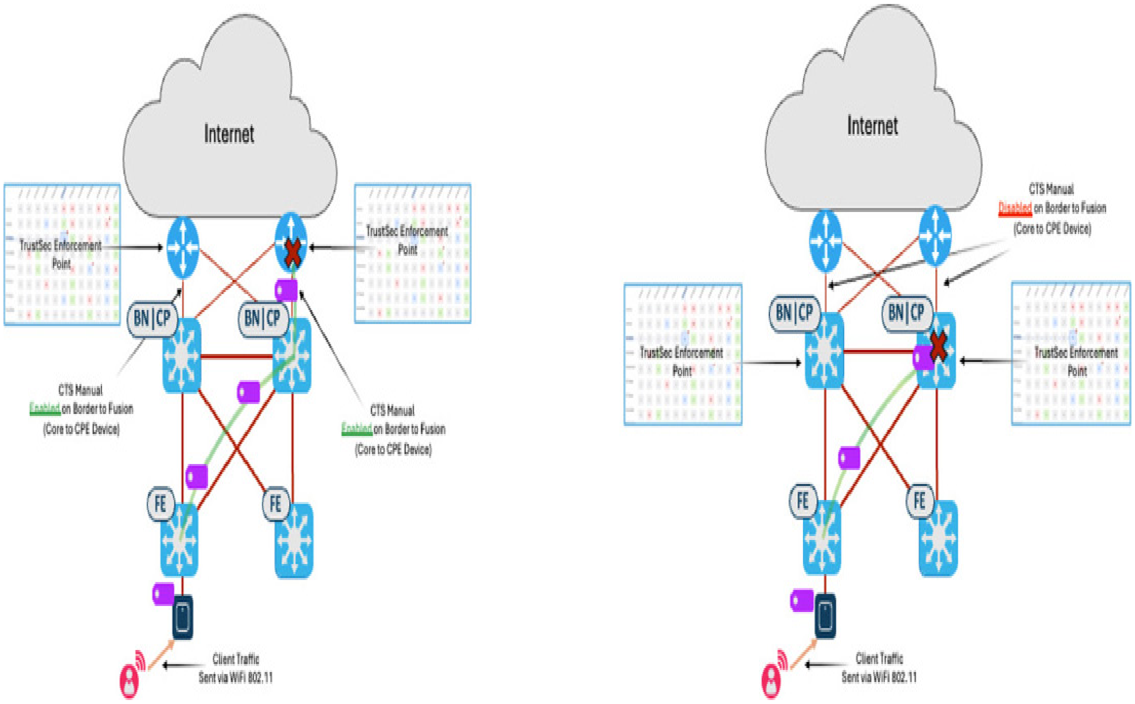
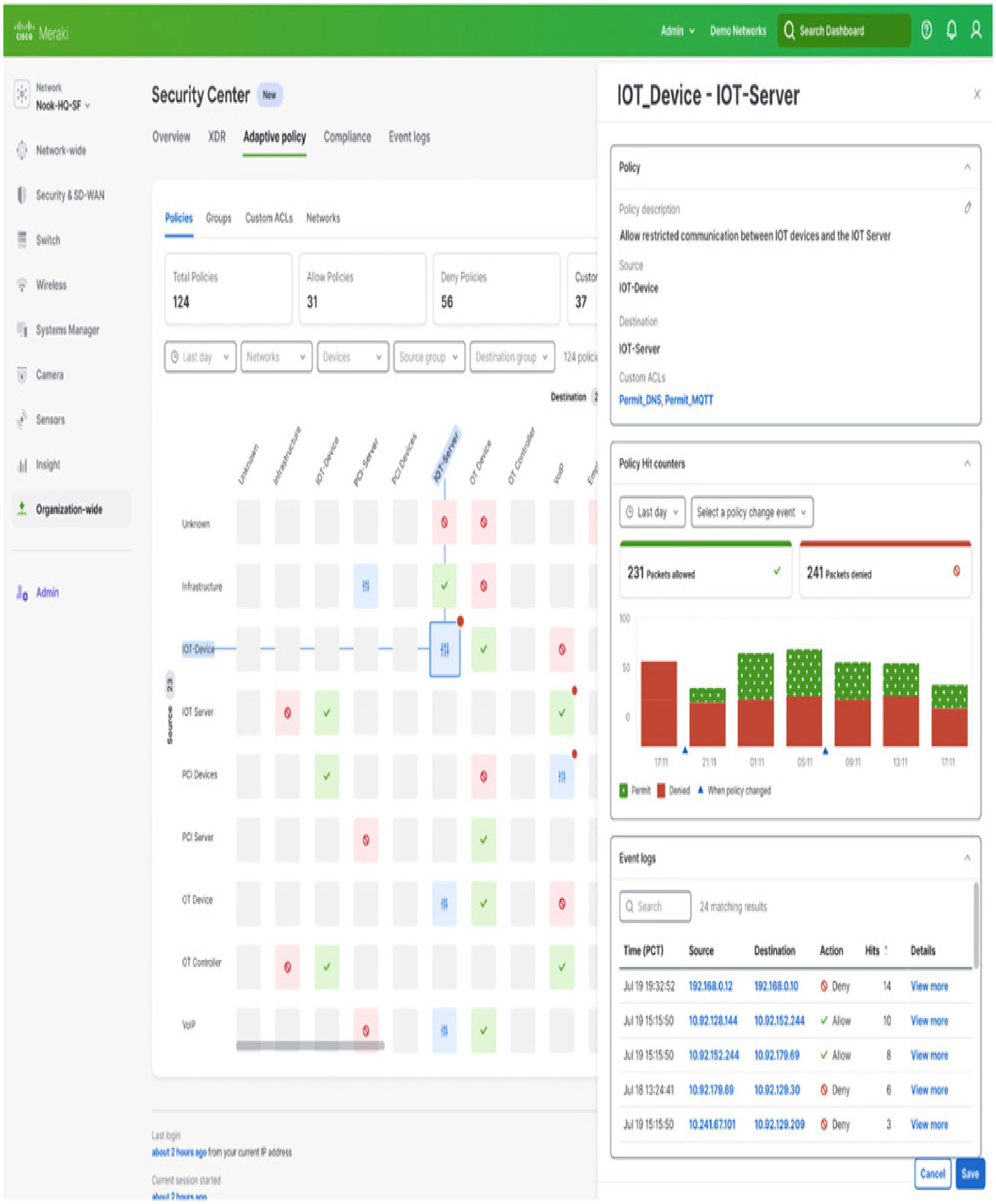

Internet
TrustSec Enforcement Point
TrustSec Enforcement Point
BN|CP
BN|CP
CTS Manual Enabled on Border to Fusion (Core to CPE Device)
CTS Manual Enabled on Border to Fusion (Core to CPE Device)
FE
FE
Client Traffic Sent via WiFi 802.11
Internet
CTS Manual Disabled on Border to Fusion (Core to CPE Device)
BN|CP
BN|CP
TrustSec Enforcement Point
TrustSec Enforcement Point
FE
FE
Client Traffic Sent via WiFi 802.11

Figure 13-22 *Enforcement Points in TrustSec*

As depicted in the figure, depending on whether the configuration of the architecture is with or without the configuration of inline tagging for TrustSec can result in a shift of the point of enforcement within the architecture.

Depending on the platform in use, different monitoring capabilities exist. Figure 13-23 shows how the Adaptive Policy view in the Meraki architecture provides an overview of policy hits. In Catalyst Center, Group-Based Policy Analytics can provide a similar view (on a per cluster basis).



IOT_Device - IOT-Server

Policy

Policy description

Allow restricted communication between IOT devices and the IOT Server

Source

IOT-Device

Destination

IOT-Server

Custom ACLs

Permit_DNS, Permit_MQTT

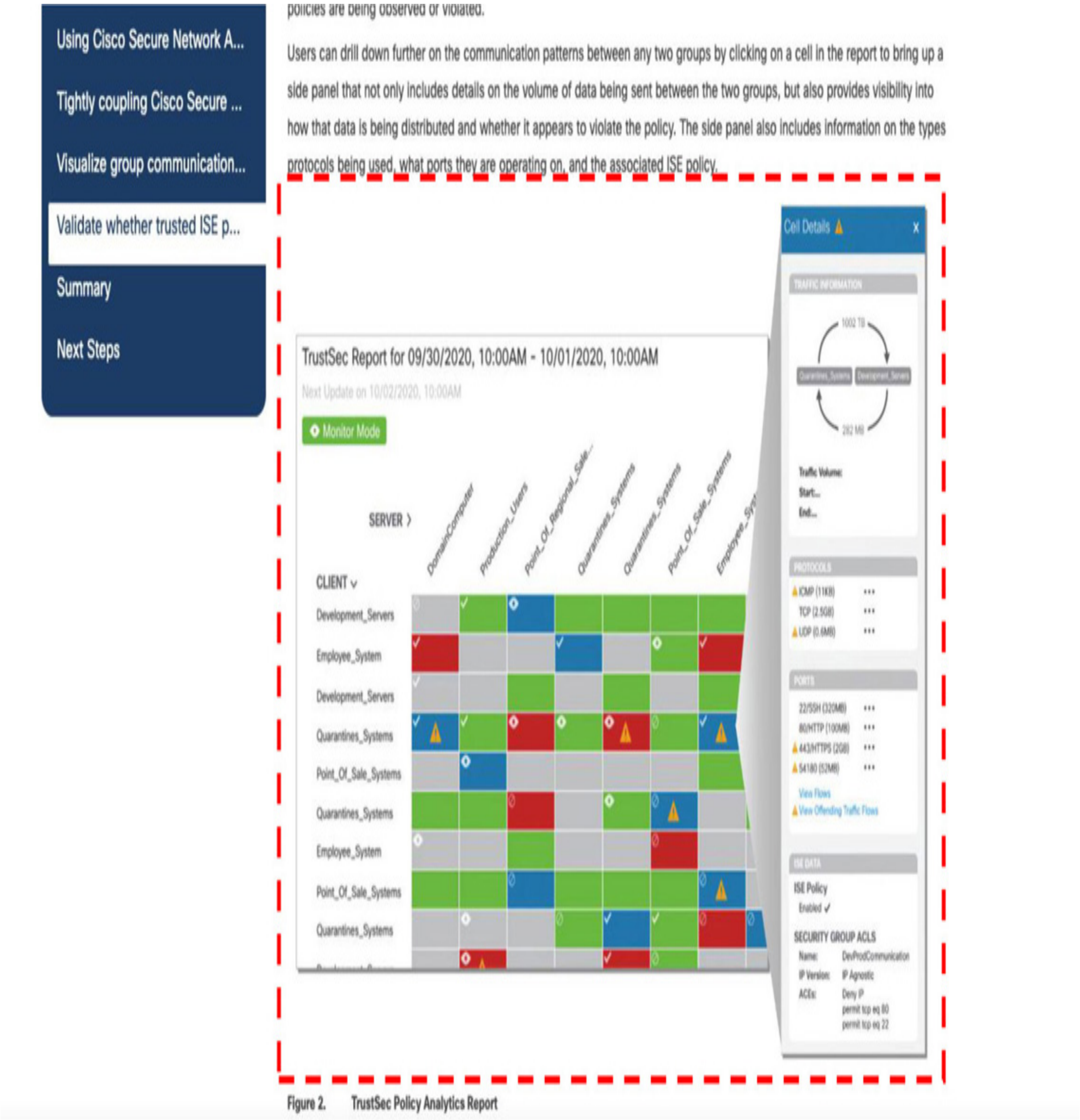
Event logs

Time (PCT)	Source	Destination	Action	Hits	Details
Jul 19 19:32:52	192.168.0.12	192.168.0.10	Deny	14	View more
Jul 19 15:15:50	10.92.128.144	10.92.152.244	✓ Allow	10	View more
Jul 19 15:15:50	10.92.152.244	10.92.179.69	✓ Allow	8	View more
Jul 18 13:24:41	10.92.179.69	10.92.129.30	Deny	6	View more
Jul 19 15:15:50	10.241.67.101	10.92.129.209	Deny	3	View more

Figure 13-23 *Adaptive Policy View in Meraki Dashboard*

A further option to get a view of policy communications in a live network is through the use of Secure Network Analytics (formerly known as Stealthwatch), which has the ability to harvest netflow data between peers, thereby showcasing the communications that are taking place between them in the form of a customizable report. An example of this data can be seen in

Figure 13-24, where the source-to-destination ruleset can be seen, with the amount of traffic that has hit the respective rule in either direction.



policies are being observed or violated.

Users can drill down further on the communication patterns between any two groups by clicking on a cell in the report to bring up a side panel that not only includes details on the volume of data being sent between the two groups, but also provides visibility into how that data is being distributed and whether it appears to violate the policy. The side panel also includes information on the types protocols being used, what ports they are operating on, and the associated ISE policy.

TrustSec Report for 09/30/2020, 10:00AM - 10/01/2020, 10:00AM

Next Update on 10/02/2020, 10:00AM

Cell Details

TRAFFIC INFORMATION

Traffic Volume:
Start...
End...

PROTOCOLS

ICMP (11KB)
TCP (2.5GB)
UDP (0.6MB)

PORTS

22/SSH (320MB)
80/HTTP (100MB)
443/HTTPS (2GB)
54180 (52MB)

ISE DATA

ISE Policy
Enabled ✓

SECURITY GROUP ACLS

Name: DevProdCommunication
IP Version: IP Agnostic
ACEs: Deny IP
permit tcp eq 80
permit tcp eq 22

Figure 2. TrustSec Policy Analytics Report

Figure 13-24 *TrustSec Policy View in Secure Network Analytics*

Using the various dashboards and data sources described earlier, an organization can actively deploy microsegmentation at scale within global network environments. Considerations should be made for how security events or anomalies that appear are correctly handled in such a domain and which events require attention. As with any security ruleset, normal background noise (or drops and passes) is commonly present in non-IOT domains; however, large swings in permits and denys typically require more focused investigations and follow-up.

Case Study: Financial Sector Customer

One of Cisco's Financial Services customers approached Cisco Professional Services to support them with a global design for their architecture. As part of this design activity, an assessment was performed to identify which global deployment options were available for deployment. During the assessment phase, the team looked toward the deployment of traditional Layer 2 network designs using Catalyst Center with hierarchically stacked components to mitigate spanning tree, a Layer 3 campus topology using the catalyst portfolio, SD-Access, and Cisco's EVPN solutions. In addition to these assessments, the Cisco and Customer teams evaluated the Meraki portfolio, considering which options were available for deployment using the cloud-based networking solution.

As part of the requirements gathering phase, a number of key and critical requirements were identified:

- The solution must be cloud-based.
- The solution must be able to be fully driven via Infrastructure as Code (IaC) methodologies.
- No graphical user interface should be permitted for any provisioning task (read-only access).
- DevOps methodologies must be followed (test, stage, prod).
- All sites must adhere to three standards (T-shirt sizes) globally.
- The solution must be able to easily interact and interoperate with other cloud-based services such as Azure EntraID.
- ITSM systems should be able to interface with the solution, to allow for a fully automated workflow to be executed based on limited user input.

In assessing and evaluating Cisco’s portfolio, and given the cloud-first focus for the deployment, the customer started looking at the key differences in the APIs to identify which solution would be the best fit. Because the customer had a limited and small operations team and had limited flexibility to travel with skilled personnel to regional sites, the Customer team came to the conclusion that the Meraki solution would be the best fit for this case.

Further defining their requirements, the team decided that the dashboard (GUI) that exists for the Meraki deployment should only be used for the purpose of operational visibility and troubleshooting, not for any configuration activities; therefore, they only permitted read-only access to it.

For read-write operations, the customer opted for introducing an IaC DevOps approach to the deployment. Maintaining all configuration files in YAML format checked into Git code repositories. Figure 13-25 shows the project workflows that were leveraged to achieve the customer requirements and converted into actionable items to execute automation workflows.

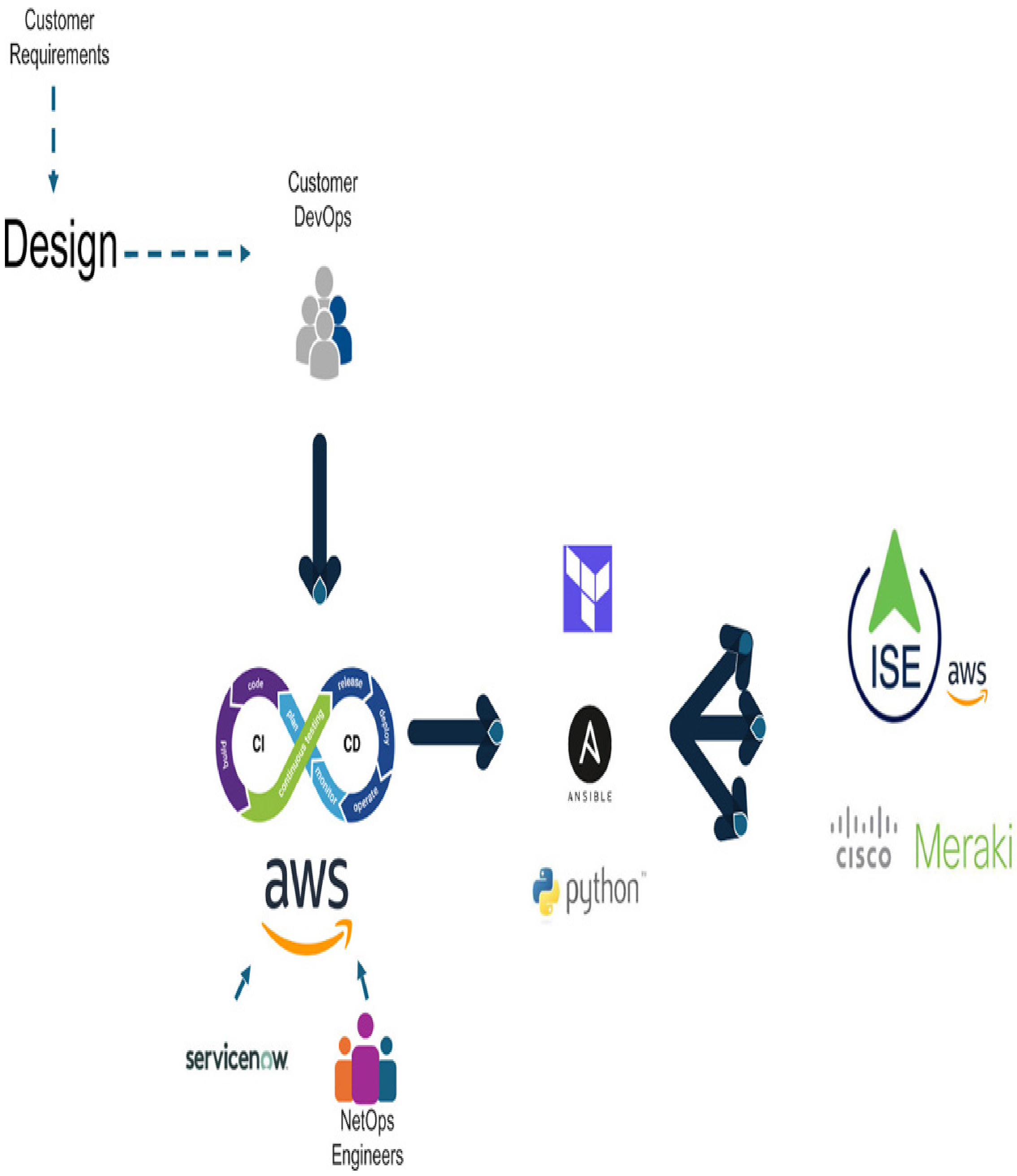


Figure 13-25 *Automation Project Workflow*

As the speed of execution with automation can be much more rapid than when leveraging human operators, prevalidation and testing are critical activities. For this reason, friendly points of presence (PoPs) were instantiated, matching each deployment t-shirt size, utilizing replica hardware to ensure that valid pre-testing was possible.

Early in the process, the agreement was made that while there may be a rich set of features available within Cisco’s portfolio and products, if those capabilities were not exposed or available via API, then they would not be approved for use. While this approach did somewhat limit the operators of the deployment to a smaller set of functionalities, the gains achieved with just working with the APIs were clear and tangible for the customer organization.

After getting the initial recipe right for day zero automation and deployment, the team added further enhancements and additions to the process to further simplify integration. These enhancements were aimed at simplifying the ability of the customer to roll out and bring up new sites and branches. To do so, they simply needed to populated a ServiceNow ITSM request with the device serial numbers they were aiming to provision, and basic details about the site including naming conventions. When the serial numbers come online, a pipeline execution would commence, resulting in the automated provisioning of a new site, and the operations team being informed via an automated bot in Webex teams of the successful completion and provisioning of the new site.

In addition to the Meraki switches, security appliances, and Wi-Fi devices undergoing provisioning, the corresponding rules required in cloud instances of ISE were also automated as part of the customer’s project, providing a complete end-to-end network setup based around automation.

Summary

In this chapter, we provided common network construct overviews, including details around operational views that may be relevant in the security context of deploying a network architecture. While we did not mention many topics in this chapter, the key tenets of any deployment construct are security and the ability to stem lateral movement of insecure or malicious devices. This chapter explored different network solutions that exist, from cloud-based to classic architectures and software-defined and automated infrastructures. In terms of which construct is the right one to choose, the answer comes down heavily to the skill set of the network operations and implementation teams, and is something that should be assessed together to ensure that the right decisions can be made within each organization.

Chapter 14. Large-Scale Software-Defined Network Deployment

In this chapter, you will learn about the following:

- Network design
- Security requirements
- Software-defined networking automation techniques
- Advantages for implementation

Introduction

In the late twentieth-century fast-food franchise, analog telephones were the extent of restaurant connectivity, with credit card transactions handled over a dial-up modem and a small number of orders received by voice phone call or fax machine during the lunch rush. Today a restaurant may take orders for dine-in or take-out by voice call, SMS, or from a smartphone application. Credit cards are verified through a cloud-based service where the kiosk in the store communicates securely over the Internet with the bank that provides card verification. Analog intercom systems at drive-up windows have been replaced with digital displays and voice-over-IP[nd]based intercoms. They still can't hear you clearly, but the screen helps to clarify what they misunderstood about your order. In the future, AI natural language processing will take the place of a human to process the incoming order. In-store ordering kiosks may be operated by a restaurant employee or they may be self-service, but either way the orders will be sent to the kitchen over the network and cashless transactions will be verified in the cloud.

In the back of the house, the systems that make the restaurant run are also connected to the network. There are screens that display orders, and there are printers that print labels. The equipment used for food preparation is networked so that the coffee maker, milk shake machine, and deep fryer can report maintenance issues and receive software updates remotely.

In the front of the house, menus are displayed on digital signs that change as the menu changes from breakfast to lunch. Music is streamed to ceiling-mounted speakers, and accent lights use power over Ethernet. A television displays streaming video content. Customers expect free Wi-Fi access while they wait for their take-out order or sit down to eat their meal.

If you are asked to design one restaurant to support all these different applications, you are probably thinking about how many switches you might need to install, or how many wireless access points are needed to cover the back of the house and the front of the house. Imagine that you need to solve the problem not once, but over 10,000 times; every restaurant needs to have the same design; and all locations need to be installed, migrated, managed, and maintained with as little downtime as possible.

In this situation, the only option is to automate the network deployment and operations end to end. In this chapter, we will examine in detail how such a massive network deployment was designed to ensure standardization and automated using the latest Infrastructure-as-Code (IaC) techniques for both day-1 deployment and day-2 operations to meet the needs of Fast Burger, one of the largest restaurant chains in the world.

Fast Burger was a huge multinational corporation, but most of the restaurants were owned by franchisees who were small-to-medium businesses (SMBs). Local Fast, LLC, was small family-run franchisee of Fast Burger that needed to migrate its 10 locations to the new system. Fred and Ethel ran a good business. Fred acted as general manager for the 10 locations, and Ethel, a certified public accountant, managed the books. Food was their largest expense, followed closely by labor and rent, with their franchise fees calculated as a percentage of sales. Each location had a manager who was responsible for day-to-day business operation and an assistant manager who was responsible for making sure the manager showed up to work each day. The rest of the employees were hourly, but they had benefits available through Fast Burger, including health insurance and a defined contribution retirement plan.

Fred was not excited about this new IT system, but Fast Burger was insistent that they adopt it, and the cost was already included in their fees. Fred enlisted Kyle, who was the manager at his flagship location, to assist with getting everything set up. Kyle got his nephew Jason, who was an engineering student at the local university, to help out as well. We will see how well Fred, Kyle, and Jason were able to adopt the new technology.

Network Design

After looking at other vendors and other hardware options from Cisco, Fast Burger chose to use Meraki equipment for its network, as shown in Figure 14-1. Meraki was the choice for several reasons: a single vendor for all the network equipment and services, a single cloud-based management platform for all the equipment, and integrated security and software-defined wide area network (SD-WAN) capabilities. Meraki has a reputation for supporting the SMB market, so it would be a good fit for their franchises as well.

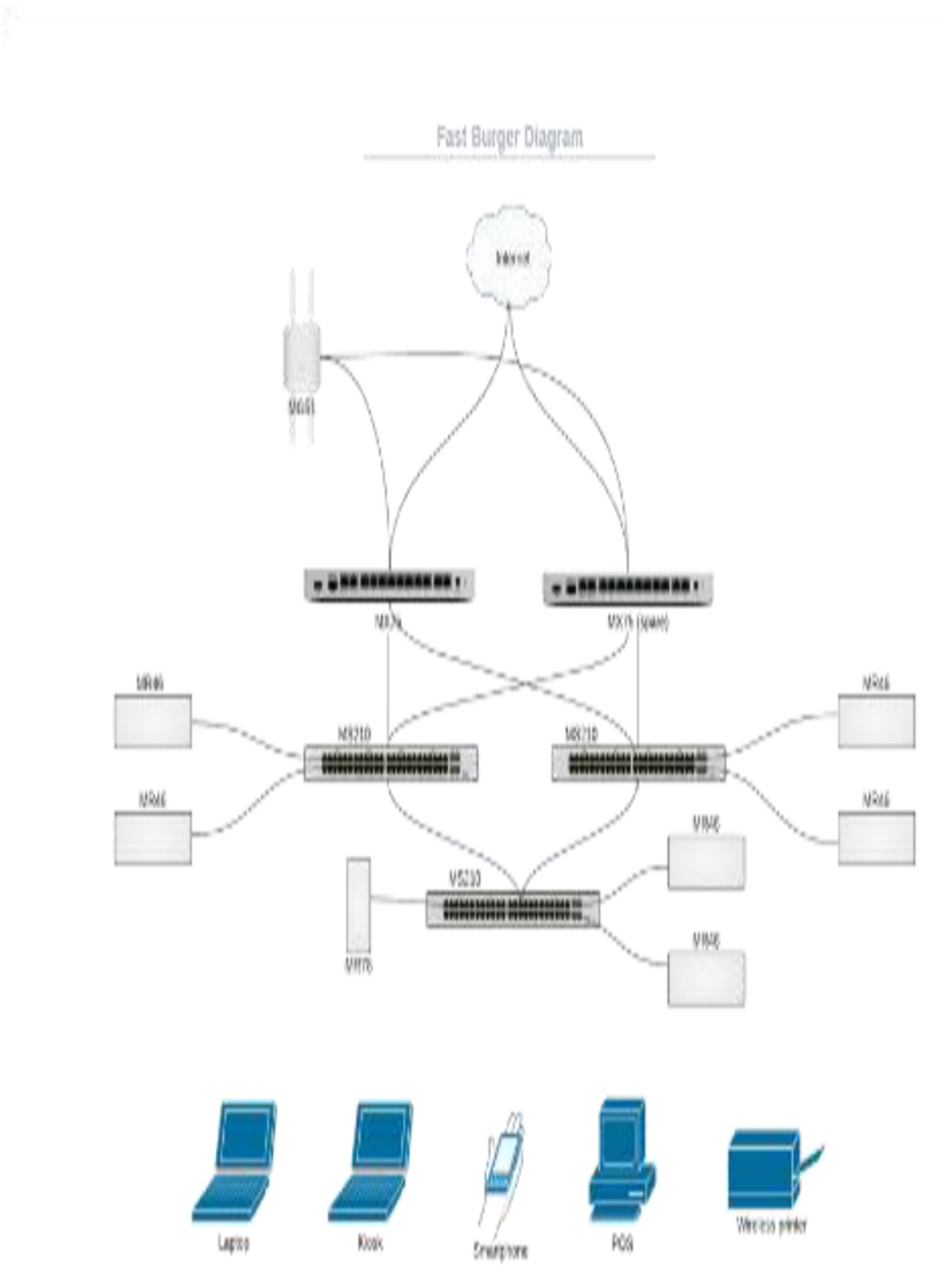


Figure 14-1 *Fast Burger Meraki-Based Network Topology*

Physical Hardware: Bill of Materials

Each restaurant required a pair of MX security gateways deployed as a redundant pair. Meraki uses the concept of a warm spare for MX redundancy, so one device stays active and the other stays in standby mode. The MX devices connected to the rest of the world using an Internet circuit for the primary uplink and an MG device with 5G wireless service for the secondary connection.

Three Meraki MS switches were needed at each restaurant. Two of the switches were cross-connected to the two MX devices. The third switch was daisy-chained to the first two switches. The switches provided wired Ethernet ports and Power over Ethernet (PoE) for all devices in the restaurant.

Each restaurant had six indoor MR access points and one outdoor MR access point providing Wi-Fi access for wireless devices including guest Wi-Fi. The access points (APs) were powered from the switch, so only one cable run was required for each device.

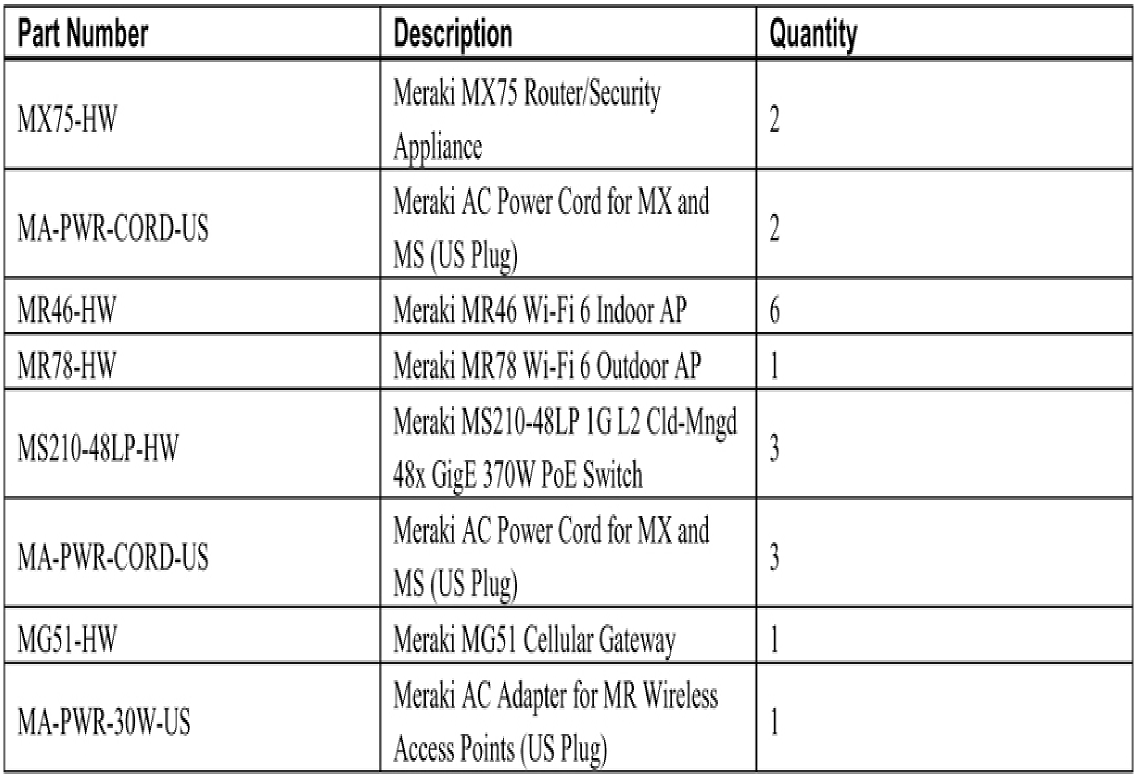
Meraki cameras and sensors were not part of the design. However, the restaurants will use power over Ethernet for their third-party cameras and other devices such as speakers and lights.

Existing cabling was reused, provided it was Category 5e or better. This also means that the positions of access points and other devices did not change when the network was migrated to Meraki.

Because of the existing standardization in the construction of Fast Burger restaurants, it was estimated that 95 percent of locations could be addressed with this basic bill of materials. Smaller locations that did not conform to the standard—for example, in open food courts, malls, or airports—could order a minimal bill of materials that just covered the back-of-the-house systems. Large nonstandard locations—for example, those that were located in renovated existing structures or that were constructed to meet certain zoning requirements in historical districts—could augment the bill of materials by ordering an add-on package of additional access points.

The majority of franchisees were expected to order the standard package, as shown in Table 14-1.

Table 14.1 *Standard Bill of Materials*



Part Number	Description	Quantity
MX75-HW	Meraki MX75 Router/Security Appliance	2
MA-PWR-CORD-US	Meraki AC Power Cord for MX and MS (US Plug)	2
MR46-HW	Meraki MR46 Wi-Fi 6 Indoor AP	6
MR78-HW	Meraki MR78 Wi-Fi 6 Outdoor AP	1
MS210-48LP-HW	Meraki MS210-48LP 1G L2 Cld-Mngd 48x GigE 370W PoE Switch	3
MA-PWR-CORD-US	Meraki AC Power Cord for MX and MS (US Plug)	3
MG51-HW	Meraki MG51 Cellular Gateway	1
MA-PWR-30W-US	Meraki AC Adapter for MR Wireless Access Points (US Plug)	1

Those franchisees whose locations did not conform to the standard could choose to order the minimal bill of materials that excluded the outdoor AP and four of the indoor APs, or for large locations the standard bill of materials with the add-on package of four additional APs.

Layer 2: Local Area Network

The Fast Burger network design relied heavily on Ethernet switching. The three switches were connected in an upside-down triangle configuration that allowed the third switch to become the root of the spanning-tree network without the need to tune spanning-tree settings. This was important because, at turn-up, the devices needed to build a loop-free tree without blocking Dashboard access for any device.

Note

> MX devices do not participate in spanning-tree, but they do forward bridge protocol data units (BPDUs).

A number VLANs were required for different applications in each restaurant:

- Point of Sale
- Internet of Things
- Physical Security
- Employee Wi-Fi
- Guest Wi-Fi

VLANs were used for macrosegmentation. Payment services devices were all placed in the point of sale VLAN. Cameras, door access, and other security devices were placed in the physical security VLAN. Connected devices in the kitchen, deep fryers, milk shake machines, and coffee makers were placed in the Internet of Things VLAN. Devices in different VLANs could only communicate with a Layer 3 connection through the MX device. There would be very limited communication between VLANs in the design because the MX devices themselves have limited throughput. If two devices needed to communicate frequently but still needed to be protected from each other, then the design placed them in the same VLAN and used adaptive policy[nd]based microsegmentation for security isolation. For example, in the guest Wi-Fi VLAN, adaptive policy assured that guest users could reach the Internet but could not communicate with each other directly.

Layer 3: Local Area Network

The MX devices served as the core of the IPv4 and IPv6 network. The infrastructure was dual stack by design with a view to future migration to IPv6. It is important to remember that the two protocols are independent of each other. Some of the devices deployed in the network were IPv4 only but would support IPv6 in the future.

Layer 3 networking within the restaurant was limited to inter-VLAN communication. Static routing was used because the MX device was the only Layer 3 device in the restaurant and did not need to exchange routes dynamically with any other device.

Secure Connect: Wide Area Network

Cisco Secure Connect is a unified, turnkey solution designed to simplify the deployment and management of Secure Access Service Edge (SASE). It integrates software-defined wide area network (SD-WAN) and Security Service Edge (SSE) to provide operational consistency across premises and cloud environments through a single Cisco Meraki dashboard. This solution supports hybrid work by securely connecting users anywhere (branch or remote) to any application (private data center, public cloud, or SaaS) with a single subscription.

Fast Burger's decision to adopt SASE was driven by its need to provide enterprise-grade security at the lowest cost to its franchisees. Because the security components in SASE, other than the local gateways, are cloud hosted, Fast Burger does not need to purchase, deploy, and maintain its own security infrastructure. Instead, the SASE vendor supplies the necessary components, and Fast Burger is able to consume enterprise-grade security as a service.

Key features include

- Integration of client-based and clientless browser-based remote worker access
- Native Cisco Meraki SD-WAN connectivity
- Comprehensive cloud-based security capabilities with zero trust network access (ZTNA)
- Enhanced traffic acquisition and unified policy

Fast Burger selected Cisco Secure Connect for its wide area network. Cisco Secure Connect with Meraki MX devices leveraged native Meraki SD-WAN integration. This integration allowed branch and data center/private cloud Meraki sites to connect securely to the Internet, SaaS, and private applications. The process involved configuring Meraki MX devices to connect to the closest Secure Connect Region, ensuring secure Internet access and private application access. The Secure Connect API is integrated with the Meraki API, which was critical for the development of automation.

Fast Burger had a number of use cases that could be met by using Cisco Secure Connect. Fast Burger required a comprehensive automation platform for unified networking and security management provided by the integration of Secure Connect into Meraki Dashboard. Secure Connect combined with Meraki Dashboard provided access to the comprehensive security features the company needed, including SASE, ZTNA, and consolidated network firewall policy and policy replication through the integration of ISE with Meraki Dashboard.

Ordering and Delivery

Fred logged in to the ServiceNow portal that Fast Burger provided and placed an order for 10 restaurant migrations. The portal generated the internal order for the equipment to be delivered to the address that was on file for Fred, which happened to be his home address. The order was sent to the warehouse, where the equipment was pulled and the serial numbers for the gear were scanned into the system. The gear shipped as one order, but the serial numbers needed to be applied to the correct restaurant in the system. The shipping documents were printed, including shipping labels and a bill of materials that showed the store-by-store allocation of equipment. The equipment was palletized and prepared for pickup by the courier.

The system also generated an email to Fred that contained setup instructions for the equipment. Fred saved the email so he could discuss it with Kyle and Jason later.

The pallets of gear arrived at Fred's house about two weeks later. Fred was out, but Ethel was there to receive the delivery. She had to back her car out of the garage and park it on the street so that a man driving a forklift could unload two huge pallets of gear into their garage. Ethel had some choice words for Fred when he arrived back later.

Security

Meraki MX devices incorporate several security features that might otherwise require a dedicated device: port-based and application-aware firewall, network monitor, and microsegmentation. Meraki devices support adaptive policy, which is Meraki's version of Cisco TrustSec.

Fast Burger chose standard 802.1x (dot1x) to manage wired and wireless authentication and assign VLAN and adaptive policy tags to ports. Cisco ISE provided both device and user authentication.

MX Security Features

In the Fast Burger environment, the Meraki MX appliance served as the cornerstone of network security, ensuring that both the business operations and customer data remained protected. The MX appliance was equipped with a suite of built-in security features that worked in tandem to provide comprehensive protection against various threats.

The Meraki MX devices included a number of security features that Fast Burger wished to take advantage of, such as Advanced Malware Protection (AMP), intrusion detection and prevention, content filtering, URL filtering, and stateful firewall capabilities.

Security in Action

Kyle was working late at Fast Burger trying to find a solution to the persistent problems with the milk shake machine. He received an email that appeared to be from a well-known food supplier. The email contained a link to what seemed like an invoice. Trusting the source, Kyle clicked on the link, which led to a phishing site designed to look like the supplier's login page. Unbeknownst to Kyle, the link also contained a payload designed to download malware onto the network.

As soon as Kyle clicked on the link, the MX appliance's URL filtering feature recognized the URL as a known phishing site and blocked access to it. This immediate action prevented Kyle from entering any sensitive information into the fake login page. Despite the original URL being blocked, the payload attempted to download malware onto the network from Github. The AMP feature scanned the file in real time and identified it as malicious. The download was immediately halted, preventing the malware from entering the network. The IDS/IPS system detected unusual traffic patterns associated with the phishing attempt. It flagged the activity as suspicious and sent a notification to the security team. The firewall rules in place ensured that only legitimate traffic was allowed, blocking attempts by the attacker to regain a connection to Kyle's device.

From Kyle's perspective, he tried to open the email attachment and got an error. He looked at the email again and realized it looked suspicious, so he deleted it. The Meraki security logs revealed that the phishing attempt was prevented, so after reviewing the incident, the security team closed the case.

ISE Integration

Fast Burger was already a Cisco ISE customer, but ISE needed to be integrated with Meraki Dashboard for each network. This process was automated on both the Meraki and the ISE side. These are the steps that were required for that integration.

Some configuration was required on ISE itself; for example, the VLANs that were used at the restaurant needed to be properly configured. This work was done as part of the initial deployment and testing in the pilot phases of Fast Burger network migration.

The remaining configuration for ISE integration happened automatically as each new restaurant was brought online. The correct radius servers were configured in the network settings in Meraki Dashboard. The access policy was configured to allow 802.1X authentication. The ports on the Meraki switches were configured to use the access policy.

Dynamic VLAN Assignment Case Study

Dynamic VLAN assignment typically involves the use of a RADIUS server to assign VLANs dynamically based on the MAC address or other criteria. The switch queries the RADIUS server to determine the appropriate VLAN for a device connected to a port. For illustration purposes, this section describes the typical message flow for a Cisco IP Phone 8665 with a voice VLAN. Some Fast Burger locations use Cisco IP phones or other similar devices, but they are not part of the network redesign because the choice of phone vendor was made by the individual franchisee.

When the IP phone is first connected to the switch, the phone and the switch start exchanging Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) messages. The first CDP/LLDP frame received from the Cisco IP phone allows the switch to recognize that a Cisco IP phone is connected to the port. This enables the switch to deliver the necessary information, such as power level and voice VLAN ID, to the phone. By default, 802.1X ensures that all traffic is dropped except for traffic needed for authentication. However, when multidomain authentication (MDA) is enabled, the switch allows CDP and LLDP traffic before authentication. This allows the phone to send and receive information regarding power requirements and the voice VLAN.

Asynchronously from the CDP/LLDP exchange, the process of authentication begins. The switch generates a RADIUS-Request message and sends it to the backend authentication server (RADIUS server). In the case of Fast Burger, ISE serves as the RADIUS server and is accessed through Secure Connect. After the Cisco IP phone is successfully authenticated via 802.1X or MAC Authentication Bypass (MAB), the AAA server sends a RADIUS-Accept message to the switch. This message includes the device-traffic-class=voice vendor-specific attribute (VSA). Upon receiving the RADIUS-Accept message, the switch authorizes the MAC address of the phone and allows it access to the voice VLAN. The switch also temporarily allows the MAC address of the phone on the data VLAN in case the phone has not yet learned the VVID. The Cisco IP phone tags all its traffic using the VVID information learned via the CDP/LLDP exchange. This tagged traffic is allowed through the switch port as a result of the authenticated MAC address of the Cisco IP phone in the voice domain. After the switch receives tagged traffic from the phone, indicating that the phone has learned the VVID, the switch no longer allows the MAC address of the phone on the data VLAN.

Adaptive Policy Microsegmentation

At Fast Burger, IoT devices such as smart appliances, point-of-sale (POS) systems, and security cameras are essential for daily operations. However, these devices can be vulnerable to cyber threats. To add an additional layer of security to these devices, Fast Burger chose to use Cisco Identity Services Engine (ISE) with Meraki MS switches and MR access points to create a microsegmentation environment using adaptive policy. This setup dynamically assigned security group tags (SGTs) to devices based on their type or other characteristics, ensuring that devices were only allowed access to the resources they needed.

Adaptive policy was also used to protect guest Wi-Fi devices from malicious activity by isolating those devices so that they can only access Internet resources and not create peer-to-peer connections within the Wi-Fi network.

Security Configuration

The original ServiceNow order also initiated the configuration in ISE. The serial numbers of the Meraki devices were used to determine the MAC addresses of the devices so that they could be provisioned in ISE ahead of time. Kyle had the ability to add and remove users from the security system using another ServiceNow portal. No new user accounts needed to be created immediately for the migration. The email that Fred received included instructions for how to provide the MAC addresses of all the devices that were not supplied by Fast Burger so that they could be added to ISE, including the video cameras, wireless speakers, and the over-the-top device that drove the television in the corner. Kyle delegated to Jason the task of climbing under tables and up on ladders to take phone pictures of the stickers on the back of each device.

Automation

Meraki Dashboard provides tools to enable the creation and replication of network designs. Templates can be created, and existing configuration can be duplicated to speed up configuration tasks. However, at the scale of Fast Burger, the built-in tools of Dashboard are not sufficient. The first limiting factor of Dashboard is that its templating tools are limited to a single organization. Fast Burger will have more than 40 organizations for its global deployment, so the automation tool must be able to replicate configuration across multiple organizations.

Meraki as Code is an Infrastructure-as-Code framework that uses a YAML-based configuration file to deploy large-scale Meraki environments. The Meraki-as-Code schema in Yamale format is derived programmatically from the OpenAPI Spec published by Meraki. A CI/CD pipeline executes a number of tools to deploy the YAML configuration. First, the configuration files are validated using a tool called *iac-validate*. iac-validate performs syntactic and semantic validation using both Yamale and a series of validation scripts written in Python. After validation, the pipeline uses Terraform to perform API calls against the Meraki Dashboard API. The Terraform components consist of a Terraform provider and a series of Terraform modules that translate the YAML files into API calls. The pipeline then executes a tool called *iac-test* that uses Robot to execute a series of test cases that verify that all configuration has been applied correctly in the Meraki Dashboard, as shown in Figure 14-2.

This solution does not require the user to understand or maintain the Terraform provider or Terraform modules. The user is responsible for creating their YAML files and maintaining their Git repository and CI/CD pipeline. Cisco maintains the Terraform provider and modules as well as the supporting utilities and schemas.

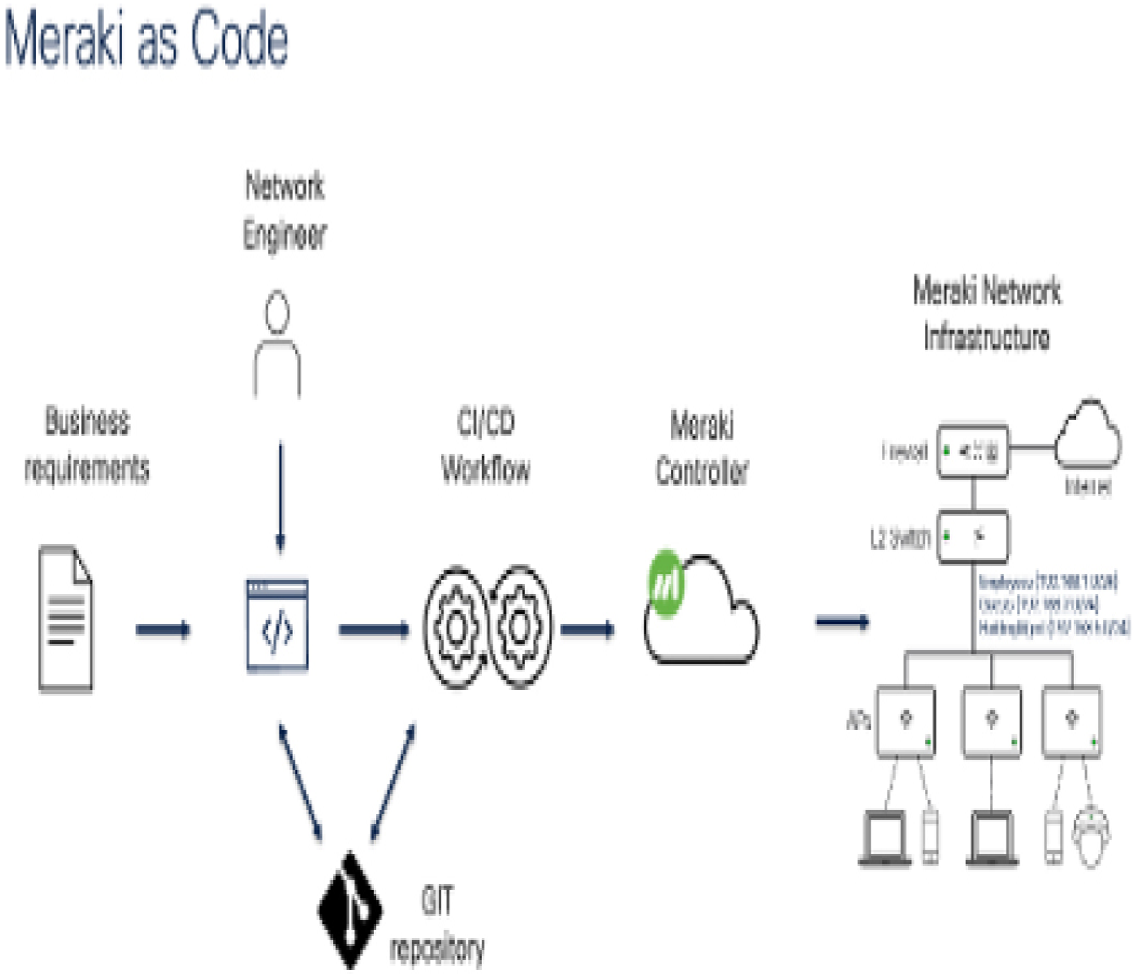


Figure 14-2 *Meraki as Code*

Meraki as Code

Fast Burger adopted Meraki as Code, an IaC framework that leverages YAML-based configuration files to deploy large-scale Meraki environments. Meraki as Code allowed Fast Burger to automate deployment of site configurations and meet its software-defined networking objectives.

Meraki as Code ensures uniform network configurations across all locations. It allows network administrators to easily replicate configurations for new locations. It reduces manual intervention and human error through automated deployments. And it includes robust validation and testing mechanisms to ensure configurations are applied correctly.

Using Git for Configuration Management

Managing configuration files efficiently and collaboratively is crucial for deploying large-scale network environments. Git, a distributed version control system, can be leveraged to handle the YAML configuration files used in the Meraki-as-Code framework, ensuring that all changes are tracked, reviewed, and deployed systematically. Here is how Fast Burger used Git for configuration management.

Cloning the master repository allows a new configuration environment to be set up. Users make changes to their local repository and then push those changes to the server. Users create templates that are reused across multiple organizations, networks, and devices. Then the users create the data files that contain organization-, network-, and device-specific data such as unique names and IP addresses. Users typically edit files in an integrated development environment (IDE) such as Microsoft Visual Studio Code (VSCode). The IDE can also be used to automate interactions with Git and the CI/CD pipeline:

```
git clone https://local.github.com/fast-burger/network-configs.
cd network-configs
```

They can create a new local branch and push the local branch to the repository:

```
git checkout main
git branch my-test-branch
git checkout my-test-branch
git add .
git commit -m "First commit"
git push origin my-test-branch
```

The YAML files are either templates that can be applied to multiple organizations, networks, or devices or data files. The data files reference which template should be used and then provide the specific values that will be populated into the API calls. A user can override a value in the template by including it in the data file. Both templates and data files are validated by iac-validate.

The configuration YAML files in Meraki as Code are stored in a structured manner. The tool is flexible enough that the exact filenames do not matter. The tool will take all of the YAML files that appear in the data directory and merge them into a single file before validation:

```
├── data/
|    ├── organization_templates.nac.yaml
|    ├── network_templates.nac.yaml
|    ├── device_templates.nac.yaml
|    ├── organizations.nac.yaml
|    ├── networks.nac.yaml
|    ├── devices.nac.yaml
```

The only requirement to create a new YAML file is to maintain the correct hierarchy within the file. For example, you could create a file named networks_wireless_ssids.yaml with the structure shown in Example 14-1. The required headers meraki>domains>organizations>networks maintain the location in the hierarchy so that the iac-validate tool can incorporate the file into the complete data structure.

Example 14-1 *YAML File*

```
meraki:
    domains:
      - name: Global
        administrator:
            name: Mr. Burger
        organizations:
          - name: Franchise-24601
            networks:
              - name: Restaurant-5309
                product_types:
                - appliance
                - camera
                - switch
                - wireless
                wireless_ssids:
                  - name: GUEST
                    enabled: true
                    auth_mode: 8021x-radius
                    available_on_all_aps: true
                    default_vlan_id: 100
                    encryption_mode: wpa-eap
                    ip_assignment_mode: Bridge mode
                    lan_isolation_enabled: false
                    mandatory_dhcp_enabled: false
                    use_vlan_tagging: true
                    splash_page: None
                    visible: true
                    wpa_encryption_mode: "WPA2 only"
                    radius_proxy_enabled: false
                    radius_testing_enabled: true
                    radius_server_timeout: 5
                    radius_server_attempts_limit: 3
                    radius_coa_enabled: true
                    radius_fallback_enabled: true
                    radius_override: true
                    radius_accounting_enabled: true
                    radius_accounting_interim_interval: 360
                    radius_attribute_for_group_policies: Filter-I
                    per_client_bandwidth_limit_down: 0
                    per_client_bandwidth_limit_up: 0
                    per_ssid_bandwidth_limit_down: 0
                    per_ssid_bandwidth_limit_up: 0
```

As users create YAML files, Git allows them to perform version control tasks. Commit changes regularly to maintain a history of modifications:

```
git add .
git commit -m "Initial commit of network configuration YAML fil
```

Users can also use branches to manage different versions or features:

```
git checkout -b feature/new-configuration
```

Multiple team members can collaborate on YAML file creation by combining forking and branching with pull requests and merge requests. They can use PRs or MRs to review and discuss changes before merging them into the main branch:

```
git push origin feature/new-configuration
```

In addition, they can integrate the Git repository with a CI/CD pipeline for automated validation, deployment, and testing:

- **Validation Stage:** iac-validate checks the YAML files for syntax and semantic correctness. This tool utilizes Yamale and custom Python scripts to ensure the configurations are error-free and adhere to the defined schema.

- **Deployment Stage:** Terraform deploys the validated configuration files to the Meraki Dashboard. This task involves using a Terraform provider developed by Meraki and a series of Terraform modules specifically designed for Meraki as Code. OpenTofu is supported as an open-source alternative to Terraform.

- **Testing Stage:** iac-test verifies that configurations are correctly applied. Robot Framework executes a series of test cases. These tests verify that all configurations are correctly applied in the Meraki Dashboard.

Users also can use issue tracking features of platforms like GitHub or GitLab to manage and prioritize changes and updates. Plus, they can maintain documentation within the repository to provide context and instructions for the configurations.

CI/CD Pipeline

Instead of executing each tool individually in a local shell, Meraki as Code uses a CI/CD pipeline to execute all of the components in temporary containers. The following list, including Examples 14-2 through 14-7, illustrates how the Jenkins CI/CD pipeline might look for validating, deploying, and testing the YAML configuration files:

1. **Setup Stage:**

   In this stage, the environment is prepared by initializing Terraform and installing the iac-validate tool for configuration validation (see Example 14-2).

Example 14-2 *Setup Stage*

```
stage('Setup') {
    steps {
        sh 'rm -rf .terraform terraform.tfstate terraform.tfstate
        sh 'mkdir -p .terraform.d'
        sh 'terraform init -input=false --upgrade' // Initializes
        sh "pip install --target=${env.WORKSPACE} --upgrade iac-v
    }
}
```

2. **Validate Stage:**

   The configuration files are validated to ensure they are properly formatted and syntactically correct (see Example 14-3).

Example 14-3 *Validate Stage*

```
stage('Validate') {
    steps {
        sh 'set -o pipefail && terraform fmt -check |& tee fmt_ou
        sh 'set -o pipefail && iac-validate data/ -v DEBUG |& tee
    }
}
```

3. **Plan Stage:**

   A Terraform plan is generated to provide a preview of the changes that will be made. This plan is then archived for future reference (see Example 14-4).

Example 14-4 *Plan Stage*

```
stage('Plan') {
    steps {
        sh 'terraform plan -out=plan.tfplan -input=false' // Crea
        sh 'terraform show -no-color plan.tfplan > plan.txt'
        sh 'terraform show -json plan.tfplan > plan.json'
        sh 'python3 .ci/github-comment.py' // Posts plan results
        archiveArtifacts 'plan.*' // Archives plan artifacts
    }
}
```

4. **Deploy Stage:**

   The approved changes are applied to the Meraki environment using Terraform (see Example 14-5).

Example 14-5 *Deploy Stage*

```
stage('Deploy') {
    steps {
        sh 'terraform apply -input=false -auto-approve' // Applie
    }
}
```

5. **Test Stage:**

After deployment, the changes are verified through various tests, including idempotency and integration tests (see Example 14-6).

Example 14-6 *Test Stage*

```
stage('Test') {
    when {
        branch 'master'
    }
    parallel {
        stage('Test Idempotency') {
            steps {
                sh 'terraform plan -input=false -detailed-exitcod
            }
        }
        stage('Test Integration') {
            steps {
                sh 'set -o pipefail && iac-test -d ./data -d ./de
./tests/results/meraki |& tee test_output.txt'
            }
            post {
                always {
                    archiveArtifacts 'tests/results/meraki/log.xh
tests/results/meraki/report.xhtml, tests/results/meraki/xunit.xml
                    junit 'tests/results/meraki/xunit.xml'
                }
            }
        }
    }
}
```

6. **Post-Build Actions:**

After the deployment, notifications are sent about the build status, and the workspace is cleaned up to remove any temporary files (see

Example 14-7).

Example 14-7 *Post-Build Actions*

```
post {
    always {
        sh "BUILD_STATUS=${currentBuild.currentResult} python3 .c
Webex notification
        sh 'rm -rf plan.* *.txt tests/results' // Cleans up artif
    }
}
```

Fast Burger Automation

After reviewing the available option for automation, Fast Burger decided to adopt the IaC methodology. Critical to its automation design was the integration of ServiceNow into the overall design. Franchisees would interact with ServiceNow through a web portal that would allow them to perform all the tasks required to order, deploy, and maintain their new Meraki environment.

Fast Burger's ServiceNow integration automatically generated the templates and other YAML files that were required for each location. The files were populated with serial numbers, IP addresses, standardized names, and other unique fields.

The files were transferred to a Git repository hosted on a private Github server with access limited to the members of the team who were responsible for preparing and maintaining the configurations. As soon as the files were in place, the CI/CD pipeline would execute.

The franchisees did not have direct access to the Git repository. Kyle's interaction with the system was entirely driven from ServiceNow. Locally unique settings like setting the VLAN and SGTs on the ports where cameras were connected were handled by opening cases in ServiceNow. Most cases could be closed automatically or with minimal intervention from Fast Burger's IT staff.

At the franchise level, Kyle knew that when he requested a port change in ServiceNow, it only took a few minutes for the change to show up on the equipment in the restaurant. The modifications to the YAML files and the execution of the CI/CD pipeline were hidden from him. In the event that something failed, his case would update with an error message. He could escalate the case to Fast Burger IT if he wasn’t able to correct the error himself.

By using Meraki as Code in addition to ServiceNow, Fast Burger ensured that all locations had consistent network configurations, reducing the risk of configuration drift and human error. The YAML-based approach allowed Fast Burger to easily replicate configurations for new locations, facilitating rapid expansion. Storing the configurations in text files rather than a database allows for troubleshooting and human intervention when problems come up. Having ServiceNow trigger the CI/CD pipeline automated the entire deployment process, from validation to testing, significantly reducing the time and effort required for manual configurations.

Implementation: Kyle and Jason Go to Fast Burger

It was a foggy autumn night when Kyle and Jason arrived at the Fast Burger restaurant, ready for their overnight mission. Both had limited knowledge about the equipment they were about to install, but they had a detailed method of procedure (MoP) document provided by Fast Burger to guide them through the process.

As they entered the restaurant, they were greeted by the whine of a bad fan bearing on the old, dusty equipment crammed into a rack in the back room. Kyle, the taller of the two, pulled out a flashlight and peered into the rack.

> "This stuff has been back here forever. Look at the dust," he said, shaking his head.

"Yeah, let's get started. The sooner we get this done, the sooner we can call it a night," replied Jason.

Following the MOP, Kyle and Jason carefully disconnected the cables from the old devices, labeling each one to ensure they could reconnect everything correctly. They unscrewed the ancient routers and switches. Kyle sneezed loudly as Jason handed him a router from the top of the rack.

"Why are we replacing all this anyway. It still works, doesn't it?" said Jason.

"Yes, it still works, but we can't get security updates for it anymore," said Kyle.

"If you're just going to throw it out, can I keep it?" asked Jason.

With the rack now empty, they began installing the new Meraki equipment. Working from the bottom up, they secured the MS switches into the rack. The two MX devices sat neatly on top of the top switch. They placed the MG cellular gateway on top of the rack so that it could get a good signal.

Next, it was time to install the MR access points in the ceiling. Using a step ladder, they had to take down the old access point and mount, install the new mount, and then the attach the MR device to the mount.

"This AP smells like french fries," said Jason as he handed the old one down to Kyle.

"I love the smell of french fries in the morning," said Kyle.

"Does it smell like victory?"

"It smells like money."

Kyle handed Jason the new MR46. Jason attached the MR to the mount, clicked the existing Ethernet cable into place, and climbed down. They worked their way around the restaurant. They also replaced the outdoor AP in the outside dining area. Because they weren't professional installers, it took them a couple of hours to do all of them, but it got easier after the first couple.

Referring to the wiring diagram in the MOP, they began reconnecting the cables. They had to spend some time untangling the mess of wires from the old gear. Ethernet cables snaked from the MS switch to the MX security appliance and MG cellular gateway. After they connected the Meraki gear, they reattached all the cables they had labeled originally. They double-checked each connection, making sure everything matched the diagram.

"All set?" Kyle asked.

"Looks good to me," Jason replied.

With a sense of anticipation, they connected the power cables on the Meraki devices. The MS, MX, and MG started to come to life, LEDs blinking in a synchronized dance. They watched the blinking lights as the devices went through their bootup sequence, first performing hardware checks and then attempting to connect to the Meraki Dashboard.

"Here goes nothing," Kyle said.

After a few minutes of staring at the lights on the devices, they were not seeing what they expected.

"Shouldn't the lights go green?" said Kyle

"Let me check the instructions. It says here that blinking orange means there is a connectivity issue." Jason began to fiddle around with the cables, trying to see what was missing.

Kyle tapped him on the shoulder and pointed to a box hanging on the wall behind the rack. "You muppet. You forgot to reconnect the cable modem."

Kyle plugged the cable modem into the uplink ports on both MX devices and power-cycled them. This time the devices went through their bootup process smoothly. As they watched, the MS devices also came up. Looking at the nearest AP, they could see that the Ethernet status light was blinking on it.

Finally, it was time to test the network. Kyle pulled out his phone and connected to the Guest_WIFI SSID. He held his breath as the connection established, and then he opened a web browser.

> "We have liftoff!" Kyle exclaimed as the Fast Burger website loaded on his phone. "I'm connected to the Internet!"
>
> "I'm glad it worked. That was a lot easier than I expected," said Jason.

Despite their limited knowledge and the apparent complexity of the task, they had successfully installed and configured the new network equipment, ensuring that Fast Burger's network was ready for the future.

They packed up their tools and prepared to leave.

> Jason turned to Kyle and said, "I'm hungry. Can we pick something up on the way home?"
>
> "Don't you want a burger?" said Kyle.
>
> "I'm vegan," said Jason.

And with that, Kyle and Jason walked out of the restaurant, leaving behind a newly upgraded, smoothly running network, ready to serve Fast Burger's customers for years to come.

Jason and Kyle are fictional characters. It is much more likely that a franchisee would pay a professional to install the equipment. The point of the narrative is that even Kyle and Jason could successfully install the equipment because of the automation systems that Fast Burger had adopted.

Summary

The Fast Burger case study illustrates how automation and security go together when building secure networks that are easy to deploy and use. The combination of Meraki Dashboard, ServiceNow, ISE, and Meraki-as-Code automation techniques allows for a customized provisioning system using mostly off-the-shelf software components.

Part 4: Cloud Security

Chapter 15. Cloud-Native Security Foundation

In this chapter, you will learn about the following:

- Cloud infrastructure security
- Key management in cloud environments
- Network security evolution and segmentation
- Navigating multicloud and hybrid cloud security
- Monitoring and logging requirements for compliance
- Emerging trends and technologies in cloud-native security

Introduction to Cloud-Native Security: A Zero Trust Perspective

With more organizations embracing cloud-native architectures as a key driver of agility, scalability, and innovation, security needs to adapt to keep pace with these dynamic environments. Central to this evolution is zero trust—a foundational security strategy that breaks from the historical perimeter-based approach. The central idea behind zero trust is that no user, system, or application can be trusted inherently, be it from within or outside the network, and hence every access request needs to be verified rigorously.

Incorporating these functions into cloud-native security practices allows for seamless integration with the zero trust model principles of continuous validation, least-privilege access, and granular segmentation. This chapter identifies cloud-native security strategies through which zero trust principles can be executed in practice. Zero trust, as a framework, forms the backbone of this entire discussion highlighting its relevance to cutting-edge cloud-native architectures—microservices, containerization, and dynamic orchestration. This chapter will explain how zero trust is the unifying, end-to-end framework to protect cloud-native environments—from understanding the shared responsibility in the cloud to deriving the value of tools for monitoring, compliance, and automated remediation.

When viewed through this lens, areas like cloud security posture management (CSPM), workload protection, and Infrastructure as Code (IaC) become context within a larger zero trust framework. Not only does this focus clarify the interplay of these concepts, but it also emphasizes their combined contribution toward the challenges associated with securing distributed and scalable systems. This chapter will specifically focus on zero trust with the goal of developing a deeper understanding of the principles behind zero trust and how those principles inform the development of a cloud-native security posture.

In the narrative to a cloud-native final destination, the transformation of security paradigms from traditional to cloud-native architectures is a notable chapter in the history of cybersecurity. Above all, the evolution is not a change in technology and processes. Instead, the evolution of how we understand and implement security strategies in the ever-growing digital reality and the birth of cloud-native was not merely a reaction to evolving security threats but a visionary shift in the way we approach digital infrastructure. Before moving forward, let’s look at some everyday terms you might have heard but not fully understood:

- **Identity and Access Management:** IAM is the backbone of cloud-native security systems that help in access control to permit what entities can access and reduce risk on the points of the cloud controller. Fine-grained access control and the least privilege reduce the incidence of data compromise and unauthorized access. On the other hand, federated IAM enables cross-domain security authentication and authorization that secure multicloud and hybrid environments.

- **Cybersecurity:** This umbrella term refers to all practices, technologies, and processes aimed to protect networks, devices, systems, and data from any form of attack, damage, or unauthorized access. Imagine, in the case of ABC Corporation, cybersecurity would be about securing all its IT systems from phishing and malware to cyber espionage. Cybersecurity also encompasses every piece of communication involved in the business, the safeguards ABC has over its assets, and the assets accessible to the world via the Internet.

- **Security in the Cloud:** Unlike security of the cloud, which refers to the protective measures managed by the cloud provider, security in the cloud pertains to how users safeguard their data and applications while utilizing cloud services. Security in the cloud considers how to protect data and applications in the cloud. It is how companies like ABC Corporation securely use cloud services. That is the way ABC ensures it controls who accesses its cloud resource, how it encrypts its data in cloud storage, and how ABC Corporation limits data and applications in the cloud from unauthorized access and prevailing cyber threats. Cloud security is the primary responsibility of the cloud user, not the service provider, thereby underpinning the shared cloud security responsibility.
- **Cloud Security:** Cloud security encompasses all measures to safeguard cloud computing environments against both external and internal threats. It includes policies, technologies, applications, and controls that protect data, applications, and the cloud infrastructure itself. For ABC Corporation, this involves securing data in transit and at rest, enforcing user permissions and access controls through authentication and authorization, and safeguarding endpoints and communication channels within the cloud environment
- **Cloud-Native Security:** This term encompasses cloud practices, tools, and strategies that ensure the safety of the application and its infrastructure purpose-built for the cloud. It depends on a dynamic, scalable, and distributed cloud-specific technology such as containers, microservices, serverless applications, and autonomous data stores. For ABC Corporation, cloud-native security is building a security firewall on the development lifecycle and inside the control infrastructure and ensuring ABC Corporation CI/CD is secure. It entails securing container orchestration systems, monitoring, and

performing data analysis, securing microservices communication using service meshes, which is an additional software layer that manages and optimizes microservices-to-microservices communication in a reliable, secure, and observable manner without requiring changes to the individual services. It facilitates this security through lightweight sidecar proxies and a central control plane, providing advanced traffic control, security, and monitoring without burdening the service code.

- **Cloud-Native Application Security**: Cloud-native application security is focused on the security of applications built on cloud-native principles. Cloud-native application security encompasses the practices and strategies to secure cloud-native applications, while Cloud-Native Application Protection Platform (CNAPP) is a unified security platform that integrates multiple capabilities, including posture management, runtime protection, and compliance checks to provide end-to-end security. CNAPP includes the security tools and practices to secure the application's lifecycle—from development to deployment and runtime. For ABC Corporation, CNAPP would indicate shifting towards DevSecOps, or making security part of the responsibility of the development, operations, and security. To do so, ABC might leverage static code analysis, dynamic code analysis, container scanning, and runtime protection measures to ensure all of its cloud-native applications are secure all the way from code inception and scaling for the cloud through production use and evolution.

Thus, just as the battlefield itself gradually evolved from fortress-based defenses to an ever-changing, dynamic theatre of war, where both the threats and defenses are in a constant state of transformation, so too did the corresponding paradigm of security evolve. Traditional security was, at its core, limited to a basic perimeter approach, a fortress wall that was supposed to keep what was precious inside and the rest out. But cloud-native architectures, fundamentally different from traditional data centers, no longer have a perimeter; rather, they rely on segmentation from the inside to create security barriers. Zero trust offers another explicit state of importance because it is not just a strategy for modernizing a user's current security implementation; it is a foundational approach that understands breach and examines every session, regardless of whether the user is found in the headquarters or a coffee shop.

From Cloud Infrastructure to Cloud Native: An Introduction to Cloud-Native Architectures

Cloud-native architectures are application architectures designed to run in cloud environments on which the flexible, scalable, and on-demand nature of the cloud is leveraged. Cloud-native architecture allows organizations to build and operate scalable applications in other modern, dynamic environments such as public, private, and hybrid clouds. We will explore the fundamental principles of cloud-native architecture in this section, with additional insights from the Cloud Native Computing Foundation (CNCF) and other industry thought leaders.

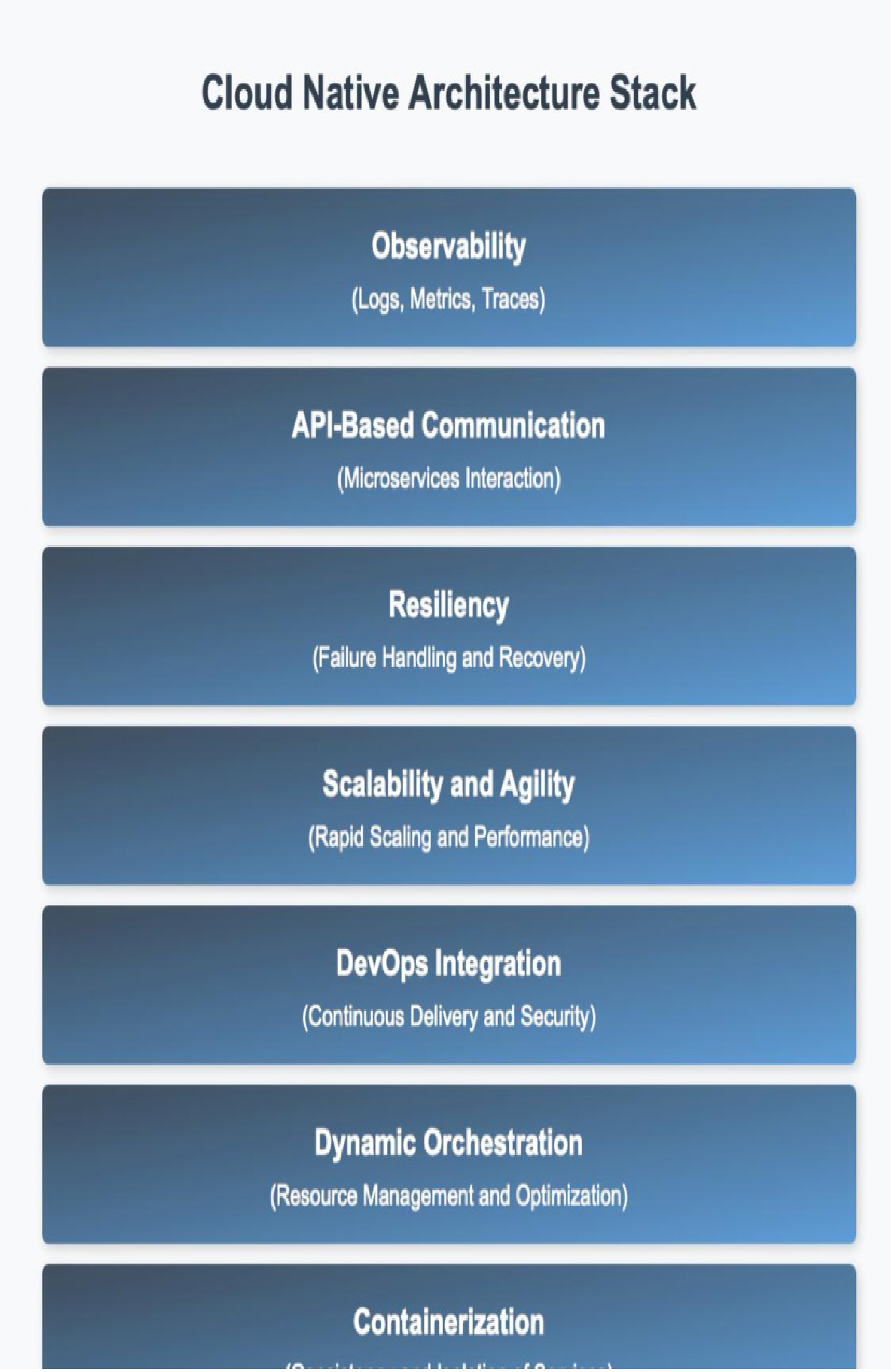
The transition from traditional cloud infrastructure to cloud-native architectures is a large shift in the way organizations handle applications from design to deployment to management. The change is not only technological but also cultural, reshaping the way teams move and how security is incorporated throughout the application lifecycle. At its basis of cloud-native architecture, there are the capabilities to change at a faster pace and tackle scale and resilience while working in today’s challenging cybersecurity.

These capabilities include cloud-native technologies, which are not an evolution of cloud but a rethinking of how applications are created and deployed. Unlike the classical and widely used “lift-and-shift” method—like hosting applications on AWS EC2 instances or Azure virtual machines, which transfer existing applications without much change, cloud-native development starts from scratch and is designed for the respective cloud environment. This enables organization to take advantage of modern cloud-native services (such as AWS Lambda, Azure Functions, or Google Cloud Run) taking advantage of cloud computing scalability, flexibility, and efficiency.

Characteristics of Cloud-Native Architectures: What Makes It Different?

Cloud-native architecture facilitates agility, scalability, resilience, and security using several guiding principles and technologies. These architectures leverage microservices for modularity, containerization for consistency, and dynamic orchestration for efficient resource allocation. For example, if you integrate DevOps practices, you will ensure continuous delivery along with security. These architectures are also designed with security and privacy by default, enabling them to handle the complexities and challenges of the modern digital environment.

The visualized stack in Figure 15-1 highlights how each layer builds on the previous one, creating a robust, secure, and efficient cloud-native architecture.



Cloud Native Architecture Stack

Observability

(Logs, Metrics, Traces)

API-Based Communication

(Microservices Interaction)

Resiliency

(Failure Handling and Recovery)

Scalability and Agility

(Rapid Scaling and Performance)

DevOps Integration

(Continuous Delivery and Security)

Dynamic Orchestration

(Resource Management and Optimization)

Containerization

(Consistency and Isolation of Services)

(Consistency and Isolation of Services)
Microservices Architecture
(Modularity and Independent Services)
Cloud-Native Foundation
(Agility, Scalability, Resilience, Security)

Figure 15-1 *Comparison Table—Cloud-Native Architecture Stack Layers*

Each layer, from bottom to top, operates as follows:

- **Cloud-Native Foundation:** This base layer represents the core principles of cloud-native architecture—agility, scalability, resilience, and security.
- **Microservices Architecture:** This layer focuses on modular and independent microservices features that facilitate the development, deployment, and scaling process.
- **Containerization:** Just above microservices, this layer ensures that each service is constantly isolated, and all of the necessary dependencies required run together so they can be deployed and scaled efficiently.
- **Dynamic Orchestration:** This layer leverages orchestration tools such as Kubernetes for dynamic deployment, scaling, and operation of containers, optimizing resource utilization.
- **DevOps Integration:** This layer integrates DevOps practices, ensuring continuous delivery and security are embedded into the development process, supporting rapid and secure updates.
- **Scalability and Agility:** Leveraging cloud infrastructure, this layer provides the capability to scale applications quickly and efficiently, maintaining performance and minimizing data inconsistencies.

- **Resiliency:** This layer focuses on building applications that can handle failures gracefully and recover quickly, ensuring continuous availability.
- **API-Based Communication:** This layer facilitates interaction between microservices through APIs, promoting loose coupling and flexibility.
- **Observability:** The top layer encompasses tools for tracking logs, metrics, and traces, providing crucial insights for monitoring, troubleshooting, and ensuring the reliable operation of applications.

Foundations of Cloud-Native Architectures

Moving to cloud-native technologies requires a deep understanding of its foundational elements—something essential for architects, developers, and security professionals. Such a journey embraces the core traits of cloud-native architectures, which we will cover briefly in the next section.

Containerization: The Building Blocks

The structure of cloud native is built on containers, upon which all other cloud-native components are built. Containers bundle application’s code, configuration, and dependencies into one singular self-contained, portable unit. The container abstraction allows the same software to run in development and production. Containers, being that they run on a shared kernel with the host system, run much faster than a virtual machine.

Docker, Containerd, and rkt are some top container technologies that enable a team to create, deploy, and run containers. Docker, in particular, has become synonymous with containers and is well known for its ability to simplify the process of developing, deploying, and running applications. On the other hand, other technologies, such as Containerd and rkt, are container runtimes used for the same processes.

Microservices Architecture: The Core of Modularity

Microservices architecture is the cornerstone of cloud-native applications. This approach involves breaking down applications into smaller independent services, each built around a single business function, which is completely different from monolithic applications' architecture. These microservices use lightweight protocols like HTTP to communicate, increasing both modularity and reliability. It's not merely a matter of dividing an application, but of allowing lower production iteration times, higher scalability, and better fault containment.

Examples of those tools are Spring Boot and gRPC. They help create stand-alone, production-grade Spring-based microservices. The Spring framework gives a powerful programming and configuration model for modern Java-based enterprise applications because it is an application-level infrastructure. This enables development teams to focus on business logic rather than get mired in details of the deployment.

gRPC is a high-performance, open-source universal RPC framework that benefits service connectivity within and across data centers. Originating at Google and part of the Cloud Native Computing Foundation ecosystem, gRPC allows for efficient communication with features like built-in load balancing, tracing, health checking, and authentication. Perfect for cloud-native platforms where performance and security are key, its bidirectional secure data streaming and built-in handling for authentication and error management fit the bill.

For example, ABC Corporation is an organization with multiple microservices within its application, each secured and monitored independently. This microservices-based architecture allows for more granular security measures, meaning that if one microservice is compromised, the impact is limited to that microservice, unlike a monolithic application where a compromise could be devastating. Service mesh technologies (such as Istio) offer an additional security layer for managing the flow in-between services through the strict enforcement of access policies and thus the end-to-end encryption.

Dynamic Orchestration and Management

Creating lifecycle management for containerized applications requires dynamic orchestration. Kubernetes, which is the de facto standard in container orchestration platforms, excels in automating the deployment, scaling, and self-healing of applications. Its robust and flexible nature allows the applications to run perfectly well and handle changing loads and demands.

Similar orchestration tools such as Docker Swarm and Apache Mesos provide extensive functionality as well. Docker Swarm is famous due to its simplicity and integration with Docker's ecosystem; it is easy to set it up and manage. The two data processing frameworks differ in terms of orchestration, however. In traditional computing, syntax-based orchestration like Apache Mesos provides a more versatile and intricate orchestration solution. This means that Apache Mesos supports various services along with multiple scheduling, high availability, scalability, and dynamic optimization.

To illustrate, Kubernetes orchestrates the deployment of containerized applications by managing the containers and services that comprise an application. It ensures they are efficiently deployed, managed, and scaled across clusters. This automation covers the entire lifecycle of containers, ensuring they function as intended and scale up or down based on real-time demand. This orchestration is crucial for maintaining the efficiency and reliability of modern, distributed applications.

For ABC Corporation, adopting Kubernetes means its applications can automatically handle failover, scaling, and updates without manual intervention. This not only reduces the operational burden but also enhances the resilience and agility of the application infrastructure.

Integrating DevOps and DevSecOps

DevOps philosophies also encourage the integration of development, operation, and security operations. Since the approach is collaborative, it seeks to maximize speed, automate processes, and integrate security from the onset. The approach also fosters an environment of continuous integration and continuous delivery while monitoring security continuously.

It speeds up the process and makes it reliable while enhancing security. Such a collaborative approach aims to optimize speed, automate processes, and incorporate security right from the beginning. They promote continuous integration and delivery (CI/CD), as well as continuous security monitoring. This results in improved speed and reliability, plus enhanced security.

These principles always manifested themselves in a broad range of tools. Jenkins is mainly utilized for provisioning/building jobs and running software testing pipelines. GitLab CI/CD works natively with GitLab because it offers a complete DevOps lifecycle right from your solutions. Aqua Security, Snyk, and Cisco's offers for Cloud Native Application Security are commonly used for security testing and compliance integrated in DevOps workflows.

At ABC Corporation, security is seen as an integral part of the DevOps process with security considerations built in from day one. Snyk can automatically scan code repositories for vulnerabilities during the development phase, but once a container is built and deployed from that vulnerable code, it needs to be protected. Aqua Security scans containers proactively before being pushed out to production. This gives continuity between design, deployment, and runtime, so ensuring security of cloud-native applications becomes automatic.

CNAPP solutions stand out by offering a comprehensive, unified security platform that comes close to offering true runtime to CI/CD pipeline security with real-time vulnerability detection across Kubernetes, serverless, and APIs. CNAPP solutions provide a complete code-to-cloud security solution unlike any other tool by supporting a shift-left approach to security testing through predeployment scanning of Infrastructure as Code (IaC) templates and scripts for risk. Generating an automated software bill of materials (SBOM) with it helps to improve supply-chain security by identifying vulnerabilities in open-source dependencies. Powered by a deep-rooted commitment to open-source innovation, these solutions go above and beyond to directly incorporate community-driven advancements into its extensive security capabilities.

We will review CNAPP solutions in more detail in Chapter 16, "Cloud-Native Application Security."

Immutable Infrastructure and Scalability

Adopting the principle of immutable infrastructure enhances security and reliability. In this approach, updates are made by replacing components entirely rather than modifying the existing ones. This principle aligns with the agility and scalability fundamental to cloud-native approaches, enabling swift adaptation to change demands and significantly enhancing security and reliability. An immutable infrastructure treats entities as replaceable and disposable rather than something to be altered post-deployment. This approach necessitates the redeployment of a new instance for any change, thereby reducing inconsistencies and enhancing both reliability and security.

Various tools support this approach. One of the most well-known examples is Terraform—an instrumental tool in enabling infrastructure provisioning using code for various service providers, thereby ensuring immutability. On the other hand, Packer by HashiCorp complements Terraform by creating identical machine images for multiple platforms using a single-source configuration. Both these tools together drive the implementation of immutable infrastructure in a cloud-native setting.

By making infrastructure immutable, ABC Corporation will address the challenge to appear in static addresses and instances in a conventional setting and the threat of configuration drift while minimizing unauthorized changes. With Terraform, there is a whole new approach of treating Infrastructure as Code. During the coding process, infrastructure is reviewed, version controlled, and audited in the same way as application code, which is secure and regulatory compliant.

Agility and Scalability: The Heartbeat of Cloud Native

The agility and scalability of cloud-native architectures allow organizations to adapt promptly to market needs and operational requirements. This characteristic is enhanced by cloud services enabling applications to vary resources automatically. For instance, cloud services like AWS Auto Scaling or Google Cloud AutoScaler exemplify this principle by automatically adjusting resources in response to real-time demand. Serverless computing models such as AWS Lambda and Google Cloud

Functions improve agility by enabling code to respond to events without dealing with servers, adjusting appropriately depending on demand.

Resiliency: Designing for Failure

Resiliency in a cloud-native infrastructure implies an application's capacity to recover from failure or continuously operate despite failures. This capability is facilitated by implementing strategies that prevent failovers from breaking services or by maintaining system availability.

Examples include Hystrix, a latency and fault-tolerance library by Netflix, which implements a circuit-breaker pattern to protect systems from cascading failures by providing fallback options. Alternatively, Kubernetes deployments can include self-healing mechanisms to recover from system or component failures.

API-Based Communication: Facilitating Service Interoperability

Microservices in cloud-native architectures communicate through APIs, ensuring loose coupling and service encapsulation. These implementations allow services to interoperate, establishing contracts between closely tied services. RESTful APIs utilize HTTP to facilitate communication between services due to their ease of expansion and performance. GraphQL, a sophisticated query language for APIs, allows clients to request specific data, providing efficient data querying and manipulation.

Observability: Insight into Cloud-Native Systems

Observability provides a real window into how apps are doing. Think of it as having eyes and ears on your system: You get to see performance, check if everything's working, and catch problems before users do. The industry calls this MELT (Metrics, Events, Logs, and Traces), and we use tools like Prometheus, ELK Stack, and Jaeger to keep tabs on everything, and Splunk, which excels at analyzing and visualizing log data for real-time insights. It's like having a health dashboard for your entire system, making it way easier to fix issues and keep users happy.

Security: A Foundational Pillar

Security is at the heart of cloud-native architecture and must be built into the application's lifecycle from the start. The key aspects include a zero trust security model, secure communication between services, secure secret management, and robust access control from the underlaying layers of the foundational infrastructure all the way to the higher layer of the application stack.

Some notable examples are Vault, a tool used to secure and store secrets, providing access for your applications to internal tasks such as tokens, passwords, certificates, and API keys. Additionally, a service mesh such as Istio provides underlayers of security by automatically encrypting service-to-service communication within the Kubernetes environment by using mutual TLS. Kubernetes has an authorization module that uses built-in roles and role-based access control (RBAC), providing a way to restrict and grant actions or capabilities to the Kubernetes API and resources, which implement access and security via the least privilege.

Building a Comprehensive Cloud-Native Security Strategy

Security remains the foundation of application deployment: Multilevel security, beginning with zero trust practices and continuing with secret management and secure communications, protects the application from threats and allows you to consider microservices architecture secure and privacy-friendly. Organizations that embody these characteristics can unlock the full potential of cloud-native technologies to empower themselves to innovate better and more efficiently. Ultimately, these guiding principles ensure the systems are available and scalable but also reliable and secure, rendering them fit for purpose in the new digital era.

As a result, shifting to cloud-native environments means that the handling of security has to be altered to meet the new requirements. Instead of functioning as an additional layer, security becomes an integral part of the architecture. Combined with the advantages in flexibility and scalability offered by cloud nativity, it can lead to a more secure situation:

- **Shift-Left Security:** Developers can secure weaknesses faster than ever before because protection is implemented earlier in the application development cycle.

- **Zero Trust Architecture:** CNAs' distributed and dynamic essence fits well with the concept of zero trust, where no item is trusted and every request to the stuff is verified and validated before access is granted.
- **Automated Security Policies:** Automate policy enforcement across the CI/CD pipeline and runtime environments to ensure consistent compliance and security. This reduces human error and accelerates security responses to new threats.
- **Enhanced Observability:** Developers can secure weaknesses faster than ever before because protection is implemented earlier in the application development cycle.

For a firm like ABC Corporation that wants to switch to cloud-native infrastructures, these ideas need to be implemented and revived at each stage of the process. This goal can be accomplished while constructing systems that are more robust, adaptable, efficient, and intrinsically secure. Only by building and integrating these ideas and methods into a unified approach to cloud-native security can digital assets prepare to thrive in a cloud-native environment. Zero trust should be seen as a process that evolves as the company grows and modifications occur.

Core Principles of Cloud-Native Architectures and Security

The core principles of cloud-native architectures and security encompass several foundational concepts that enhance the overall security posture of digital infrastructures. The first one, as mentioned previously, is *immutability*. Once a component is deployed in a cloud-native environment, it does not change. This means it cannot be altered to install unauthorized modifications. Through immutable deployments, cloud-native-certified container technologies prevent hackers from making unauthorized changes that could compromise a system. Immutable deployments also ensure that users are always using the latest known version, eliminating the need to switch between different versions of the system.

Another aspect is the concept of *Declarative APIs*, which allow application designers to explicitly state the desired resource state, enabling the system to automatically handle the implementation of the specification regardless of the current cluster state. This eliminates the need for complex controllers to match the resource state with the specification. *Observability*, on the other hand, involves tools such as logging, monitoring, and tracing, which provide details about application performance and security incidents. Without logging and monitoring, platform and cloud administrators' ability to react to security threats is limited, because they are unable to observe and proactively address such threats.

Lastly is principle of *modernizing the application*, which means that transitioning from a monolithic to a cloud-native microservices architecture improves agility and comes with numerous advantages. However, it also brings significant implications for security, such as the need to ensure secure service-to-service communication and manage more complex networking topologies due to additional layers and constructs. These security concerns can be addressed using service meshes like Istio, which provide end-to-end encryption and fine-grained access control to properly secure service-to-service communication.

Why Microservices and Immutability Enhance Security

Microservices inherently reduce the potential attack surface by dividing the application into smaller, independently deployable services. Such an approach means that any security problem in a specific microservice affects only that part of the deployed application, rather than the entire system. In contrast, since monolithic applications have a single codebase and are managed as a single entity, any threat to the monolithic system potentially compromises the entire application. Furthermore, each microservice can be monitored and administered separately, allowing detection and response measures be tuned to the specific requirements of the unit to which they apply.

Additionally, microservices are immutable, which prevents unauthorized modifications to the deployed container or configurational images. If ABC Corporation utilizes immutable containers, any attempt to modify the application at runtime would be detectable and could be automatically prevented or rolled back. It could then be automatically stopped or, in the case of manual intervention, easily reversed, returning the system to its original state. Finally, if the immutability measures fail, the compromise is likely to be severe and would have been noticed relatively early. Therefore, the containers could be swiftly and easily rolled back to the last state before the compromise, in any case, to a known-good state, which can be done predictably and will necessarily also minimize downtime. Overall, these principles ensure that modern cloud-native architectures not only can support efficient, scalable, and robust operational models but their very fabric and inner workings are imbued with security.

Service-to-Service Communication vs. Traditional Centralized Firewall

As noted, within the context of cloud-native architectures, the traditional practice of utilizing centralized firewalls to monitor and manage all network traffic has been replaced by a mechanism of a more dynamic model of service-to-service communication among microservices. This new approach allows microservices to interact directly with each other and eliminates the requirement for all traffic to be steered via the same firewall. As a result, this allows the establishment of fine-grained security policies designed to address each microservice's needs and particular risks.

For example, at ABC Corporation, this setup allows deploying unique encryption and authentication procedures to safeguard sensitive data exchanges. As a result, data interception or unauthorized transmission is extremely unlikely. Additionally, service meshes, such as Istio, aid in the automatic implementation of security policies across the services, allowing organizations to deploy consistent security measures without the need for each microservice.

Conversely, while the traditional centralized firewall and overall perimeter-based protective system remain a dependable form of defense, it has significant operational constraints. Primary among them is that the firewall is a single point of failure; the entire network can be compromised if attackers manage to bypass the central firewalls. In addition, the central systems cannot repel internal threats nor efforts of lateral movements through the network. Within the setting of modern-day dynamic and decentralized cloud-native applications, handling the sheer volume of communication between the services may be overwhelming for centralized systems. In addition, the level of control that can be implemented and the specific security policies achievable with service meshes are simply not plausible within a centralized system are simply not plausible within a centralized system. The shift from centralist firewall systems to service-to-service communication in a distributed setup illustrates an epochal shift in security approaches with cloud native. It outlines how cloud-native technologies and software can support inculcating adequate measures with specific controls as required by each service.

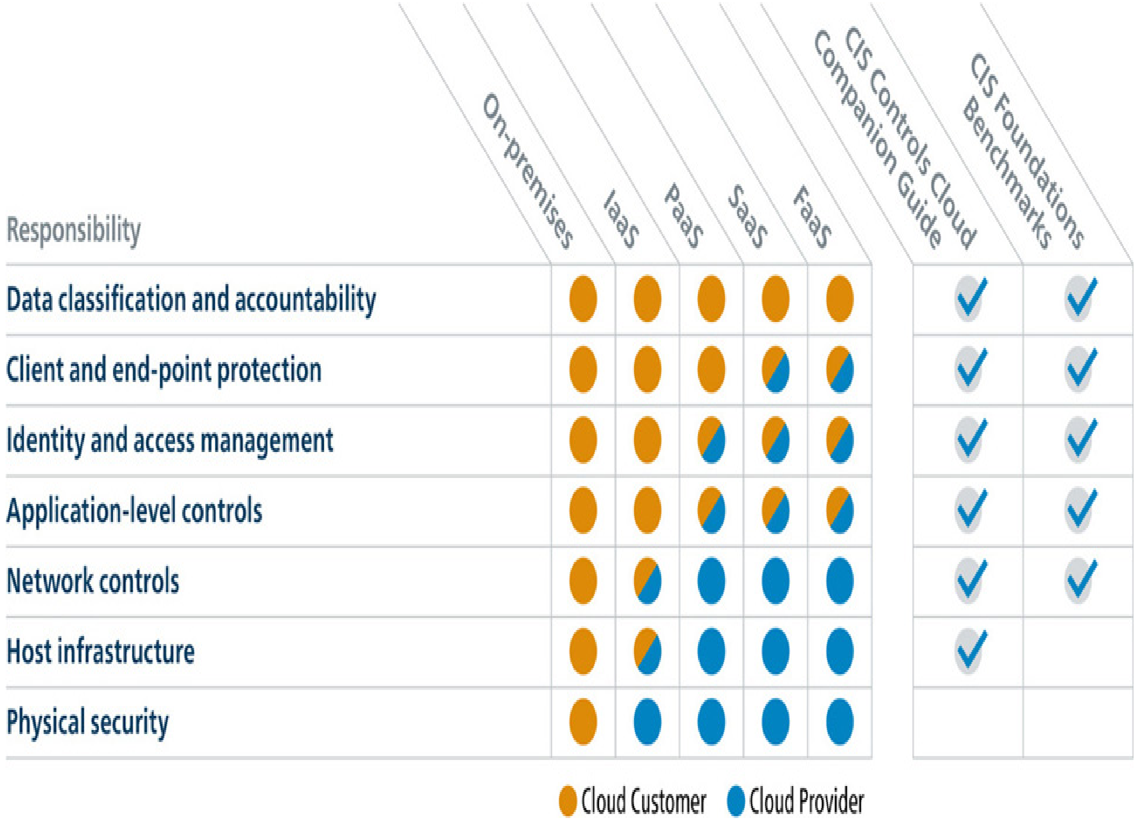
Cloud Infrastructure Security: Pillars and Practices in the Modern Cloud

One of the main issues that cloud computing is still concerned about is security, which defines the architecture and deployment of cloud-native applications. It includes the understanding of the shared responsibility model and implementation of security measures in cloud platform and software stacks. The primary approach also implies specialized data classifications, robust data protection methods, advanced data encryption solutions, and secure Internet protocols integrated into cloud architectures. Therefore, it seems reasonable to discuss security pillars in the modern cloud, outlining the key principles of the cloud-native architecture and how they affect security.

The Shared Responsibility Model: A Foundation of Cloud Security

The shared responsibility model is a key component of cloud security, because it serves as a foundation for determining the cloud providers’ and cloud users’ security responsibilities. The concept is a system of priorities that relies on an allocation of security duties between the two parties: Cloud providers are responsible for protecting the infrastructure that provides services. This encompasses the hardware, software, networking, and facilities required to maintain the cloud. At the same time, the users of these cloud services are responsible for protecting their specific data within the cloud. This includes security settings for data held within the cloud, as well as the platforms, applications, and programs they provide. The division of responsibilities is designed to enable a holistic and secure posture.

Figure 15-2 exemplifies the division of responsibilities; it is a crucial aspect of effective cloud-native security implementation because it ensures the expertise of the provider in securing the infrastructure, while allowing the customer to focus on security at the application level. Therefore, the understanding of the model is critical for users conducting operations within the cloud, as it allows them to have a clear understanding of their security responsibilities.



Responsibility	On-premises	IaaS	PaaS	SaaS	FaaS	CIS Controls Cloud Companion Guide	CIS Foundations Benchmarks
Data classification and accountability	🟠	🟠	🟠	🟠	🟠	✓	✓
Client and end-point protection	🟠	🟠	🟠	◐	◐	✓	✓
Identity and access management	🟠	🟠	◐	◐	◐	✓	✓
Application-level controls	🟠	🟠	◐	◐	◐	✓	✓
Network controls	🟠	◐	🔵	🔵	🔵	✓	✓
Host infrastructure	🟠	◐	🔵	🔵	🔵	✓	
Physical security	🟠	🔵	🔵	🔵	🔵		

🟠 Cloud Customer 🔵 Cloud Provider

Figure 15-2 *Comparison Table: The Shared Responsibility Model for Cloud Security*

** The CIS Foundations column represents a benchmark according to the CIS market standard. According to the official website by the Center for Internet Security: “The CIS Foundations Benchmarks are a part of the family of cybersecurity standards managed by the Center for Internet Security (CIS). CIS Benchmarks are consensus-based, vendor-agnostic secure configuration guidelines for the most commonly used systems and technologies.” For more information, refer to the “Foundational Cloud Security with CIS Benchmarks” (see the “References” section at the end of this chapter.).

Architectural Foundations of Cloud Security in Hyperscaler Platforms

The foundations of best security practices in a cloud environment are the building blocks that determine the robustness of a cloud security posture. These principles form the pillars upon which a cloud-based approach to securing data, managing access, and maintaining the integrity and availability of cloud services is built. Hyperscalers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, offer numerous cloud services that enable cloud-native security at scale. Some of these integrated security services are meant to secure applications and data and automate compliance checks in a cloud environment and provide identity and access management capabilities that are critical for zero trust architectures. Hyperscaler platforms, through architecture and services, enhance cloud-native security by enabling a wide range of services leveraging cloud computing’s scalability, flexibility, and resilience. They provide integrated security features that address modern applications’ broad spectrum of requirements. For example,

- **AWS:** Amazon GuardDuty secures threat detection, AWS Identity and Access Management secure access, and Amazon Inspector automates security assessment.
- **Google Cloud Platform:** Google Cloud Armor secures network, Google Cloud Identity and Access Management secures access, and Security Command Center manages security and data risk analysis in GCP services.
- **Microsoft Azure:** Azure Active Directory manages identity, Azure Policy ensures organization, and Azure Sentinel manages security information and event management (SIEM).

Unified Security Models and Identity and Access Management (IAM)

Unified security models are fully integrated into hyperscalers’ cloud environments, spanning all services from compute and storage to networking and databases. This integration allows ABC Corporation to enforce security configurations and rules consistently across its entire cloud infrastructure, ensuring a cohesive security posture.

At the core of this security architecture lies Identity and Access Management (IAM). IAM provides a structured approach to managing digital identities and access rights across systems and services. It allows organizations to enforce fine-grained permissions, ensuring that only authorized users and endpoints can access specific resources. This aligns with the principle of least privilege, minimizing the attack surface by granting the least amount of access necessary.

For example, in AWS, accessing any resource requires an associated IAM role or policy. These roles and policies explicitly define what users or services can and cannot do, ensuring precise control over resource access. At ABC Corporation, this allows for efficient permission management, reducing the risk of unauthorized changes. By implementing stringent access policies, ABC Corporation ensures that only authorized personnel can alter critical configurations, enhancing overall security.

Data Encryption and Protection

Hyperscalers offer robust data encryption capabilities to safeguard data both at rest and in transit. GCP’s Cloud Storage, for example, automatically encrypts all data before it is written to disk, with no additional actions required from the user. This ensures the confidentiality and integrity of data. Employing advanced encryption standards, such as AES, and protocols like Transport Layer Security (TLS) is crucial for securing data. These encryption methods protect data whether it’s stored in the cloud or transmitted across networks. Many cloud providers offer built-in encryption capabilities, which simplifies the process for users to secure their data without requiring in-depth cryptographic knowledge.

Effective cloud security strategies start with the classification and protection of data based on its sensitivity. Identifying the nature of data helps in applying the appropriate level of security controls. Data encryption, both at rest and in transit, is paramount in preventing unauthorized access. Cloud services like AWS Key Management Service (KMS) and Azure Key

Vault provide robust mechanisms for encryption key management, enabling secure data storage and transmission.

As mentioned, secure internet protocols like HTTPS or beast-mode with Transport Layer Security facilitate a secure channel through which data can be transmitted and is critical for protecting data in transit. In the context of cloud-native architectures, such security is further bolstered by the use of API gateways and service meshes that enable the application of tough communication protocols in the microservices. These components operate within different layers of a typical software architecture stack. The API gateway works at the application level, managing API traffic from edge level client-to-service, while the service mesh operates on the lower infrastructure level, dividing application functionality into microservices and managing internal service-to-service communication.

Additionally, adopting the *Security by Design* approach ensures that the critical network and infrastructure security measures are built right into the design of the architecture. These measures include secure coding, making use of IaC methodologies to achieve consistent and secure functions, and the principle of least privilege, which scientifically reduces access rights and attack surfaces. As shown previously, integrated practices for the network and infrastructure aspect of data security make cloud-native environments not only viable and agile but also strong to defend against and deal with cyber threats.

Network and Infrastructure Security

Network security features such as firewalls, virtual private clouds (VPCs), and subnetting provide detailed segmentation and isolation of resources, key to creating security boundaries around sensitive workloads. Foundational to public cloud security is the underlying network and infrastructure security, aimed at protecting data, applications, and services from unauthorized access and cyber threats. Several technologies and practices are securing ABC’s network resources against vulnerabilities and attacks and ensuring the secure operation of cloud deployments: network security groups, application security groups, and an additional layer of isolation at the network level (again through Vnets). *Vnets* are Microsoft Azure’s proprietary implementation of the popular concept of VPCs.

The main components of network and infrastructure security include, but are not limited to:

1. **Network Segmentation and Microsegmentation:** A network is divided into small, isolated sections. This segmentation helps to control traffic flow and prevent the expansion of hostile forces within the on-premises cloud environment. Microsegmentation offers granular control over network traffic at the workload level, enabling more precise security policies tailored to specific applications or services.
2. **Firewalls and Intrusion Detection Systems/Intrusion Prevention Systems (IDS/IPS):** Cloud-based firewalls and IDS/IPS are essential tools for monitoring and controlling incoming and outgoing network traffic based on predetermined security rules. These solutions help in detecting and preventing unauthorized access, exploits, and other malicious activities within and between logical networks and cloud resources and can be applied for both north-south and east-west traffic patterns, if implemented correctly.
3. **Secure Virtual Private Networks (VPNs) and Encryption:** Utilizing VPNs for encrypted connections between cloud resources and users ensures the secure transmission of data across the Internet. Encryption of network traffic protects sensitive information from eavesdropping and interception during transit.
4. **Zero Trust Network Access (ZTNA):** The zero trust model advocates for "never trust, always verify" as a guiding principle for network access. ZTNA solutions enforce identity verification and context-aware access policies, ensuring that users and devices are authenticated and authorized before granting access to network resources.
5. **Public and Private Cloud Connectivity:** Securely managing the connectivity between public and private clouds is vital for hybrid cloud environments. This involves the implementation of secure gateways, dedicated connections (such as AWS Direct Connect or Azure Express Route), and consistent security policies across cloud boundaries to maintain the integrity and confidentiality of data.

6. **DDoS Protection and Traffic Management:** Protecting cloud services from distributed denial-of-service (DDoS) attacks is crucial to maintaining availability and performance. Cloud providers offer DDoS mitigation tools and traffic management solutions to absorb and deflect attack traffic, safeguarding cloud resources from disruption.
7. **Regular Security Assessments and Penetration Testing:** Conducting regular security assessments and penetration testing of the network and infrastructure helps in identifying vulnerabilities and weaknesses. These evaluations enable proactive remediation of security gaps, strengthening the defense against potential attacks.
8. **Integrating Network and Infrastructure Security:** Integrating network and infrastructure security measures into the broader cloud security strategy is essential for achieving a holistic defense posture. Collaboration between security, network, and cloud teams ensures that security controls are aligned with organizational needs and cloud architectures. By leveraging cloud-native tools and services provided by cloud providers, alongside third-party solutions, organizations can build resilient, secure networks that support the dynamic nature of cloud computing.

Implementing Automated Compliance and Governance

Hyperscalers provide automated compliance toolsets and continued monitoring tools to ensure that the environment remains compliant with a regulated standard, discovering all noncompliant resources (and even sometimes auto-remediating). Fortifying cloud security from top-to-bottom with adherence to compliance standards (whether they be industry-specific or internal) and sound governance to protect data and solidify that trust is critical. Compliance certifications from cloud providers simplify the task of adhering to industry regulations, and governance frameworks enforce policies for resource usage, data protection, and operational controls, resulting in consistent security with cloud deployments.

For example, ABC Corporation uses AWS Config to assess, audit, and evaluate the configurations of its AWS resources in order to monitor compliance with the company's security policies as well as regulatory standards.

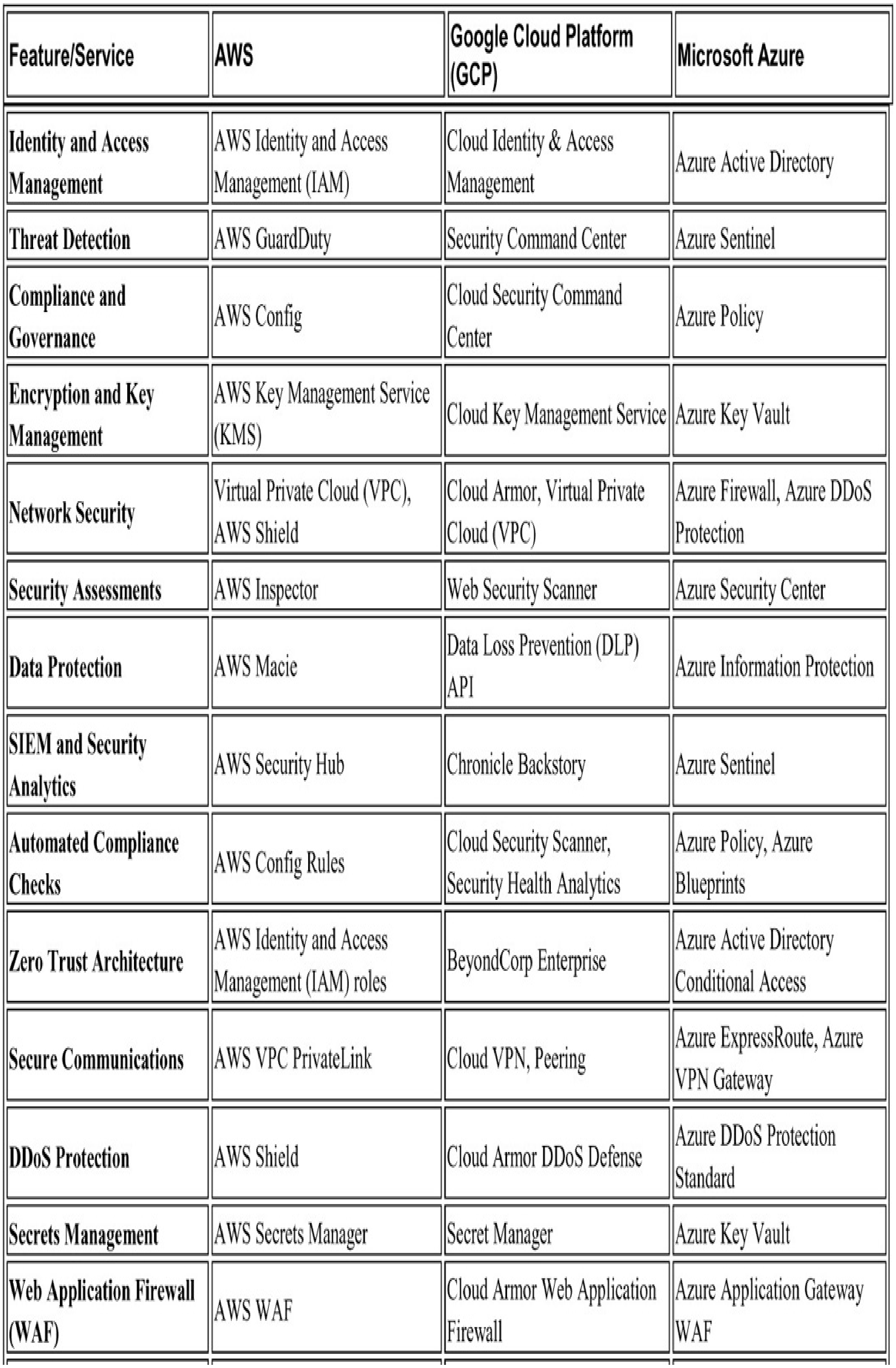
Security Monitoring: Threat Detection and Response

Monitoring and threat detection services need to be able to see through the entire stack in real time without impacting the performance of critical applications. These platforms utilize advanced analytics and machine learning to rapidly detect and respond to threats. The key to effective security for today and into the future is continuous visibility regarding potential threats and a plan of action. Cloud providers provide tools and services that can monitor, detect threats, and automatically respond to security incidents, offering real-time visibility and making it possible to identify threats far before they impact the business, thus substantially limiting their potential damage.

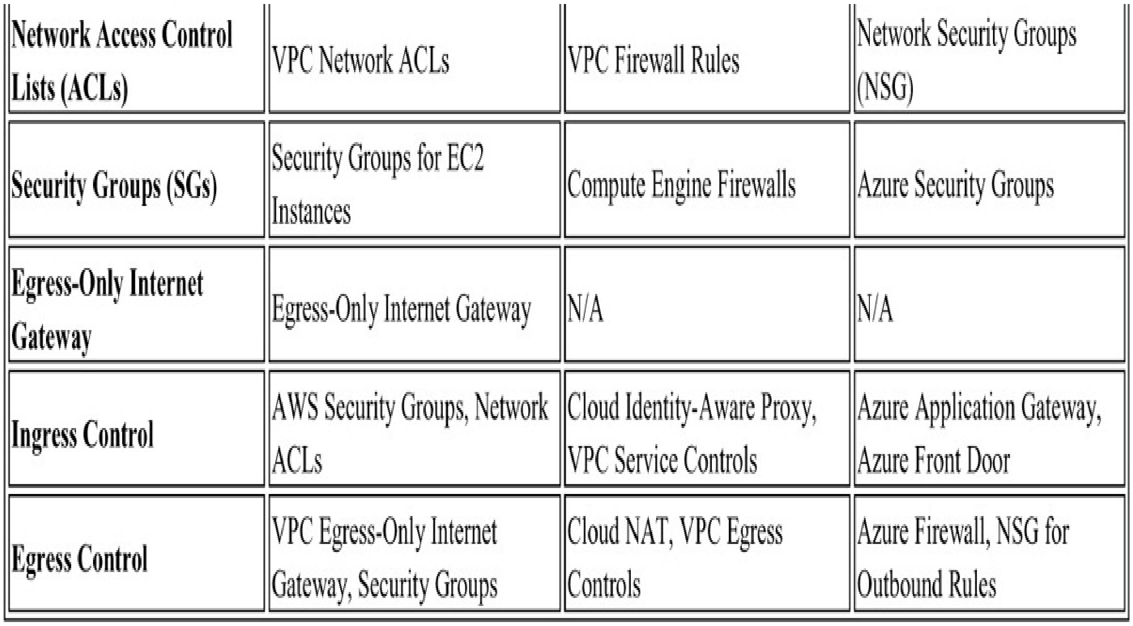
For example, the cloud-native SIEM service Azure Sentinel provides ABC Corporation with a comprehensive overview of its entire Azure deployment and employs artificial intelligence (AI) to automatically detect, investigate, and act on threats across the board.

Table 15-1 provides a clearer understanding of the network and access control services that AWS, GCP, and Azure offer to enhance security within their cloud environments.

Table 15.1 *Summary of Network and Access Control Services of All Three Hyperscalers*



Feature/Service	AWS	Google Cloud Platform (GCP)	Microsoft Azure
Identity and Access Management	AWS Identity and Access Management (IAM)	Cloud Identity & Access Management	Azure Active Directory
Threat Detection	AWS GuardDuty	Security Command Center	Azure Sentinel
Compliance and Governance	AWS Config	Cloud Security Command Center	Azure Policy
Encryption and Key Management	AWS Key Management Service (KMS)	Cloud Key Management Service	Azure Key Vault
Network Security	Virtual Private Cloud (VPC), AWS Shield	Cloud Armor, Virtual Private Cloud (VPC)	Azure Firewall, Azure DDoS Protection
Security Assessments	AWS Inspector	Web Security Scanner	Azure Security Center
Data Protection	AWS Macie	Data Loss Prevention (DLP) API	Azure Information Protection
SIEM and Security Analytics	AWS Security Hub	Chronicle Backstory	Azure Sentinel
Automated Compliance Checks	AWS Config Rules	Cloud Security Scanner, Security Health Analytics	Azure Policy, Azure Blueprints
Zero Trust Architecture	AWS Identity and Access Management (IAM) roles	BeyondCorp Enterprise	Azure Active Directory Conditional Access
Secure Communications	AWS VPC PrivateLink	Cloud VPN, Peering	Azure ExpressRoute, Azure VPN Gateway
DDoS Protection	AWS Shield	Cloud Armor DDoS Defense	Azure DDoS Protection Standard
Secrets Management	AWS Secrets Manager	Secret Manager	Azure Key Vault
Web Application Firewall (WAF)	AWS WAF	Cloud Armor Web Application Firewall	Azure Application Gateway WAF



Network Access Control Lists (ACLs)	VPC Network ACLs	VPC Firewall Rules	Network Security Groups (NSG)
Security Groups (SGs)	Security Groups for EC2 Instances	Compute Engine Firewalls	Azure Security Groups
Egress-Only Internet Gateway	Egress-Only Internet Gateway	N/A	N/A
Ingress Control	AWS Security Groups, Network ACLs	Cloud Identity-Aware Proxy, VPC Service Controls	Azure Application Gateway, Azure Front Door
Egress Control	VPC Egress-Only Internet Gateway, Security Groups	Cloud NAT, VPC Egress Controls	Azure Firewall, NSG for Outbound Rules

These hyperscalers have all built a comprehensive suite of tools that allow traffic to be meticulously examined and controlled, further fortifying the data and applications they deliver inside their infrastructure. A combination of these tools can help organizations achieve a more mature security posture, because they are based on best practices and establish benchmarks for regulatory compliance. With these concepts and services, the likes of ABC Corporation can leverage out-of-the-box security features and services intended for cloud-native applications. Hyperscalers provide the infrastructure and services needed for cloud-native computing together with security at all levels of the cloud stack.

Key Management in Cloud Environments

Alongside other characteristics, securing data and protecting privacy are a matter of managing cryptographic keys very carefully, especially with the increase of running workloads in the cloud these days. Key management, the process of handling these cryptographic keys, plays a crucial role in encryption practices, providing the backbone for securing data in the cloud. As we transition from on-premises to cloud-based infrastructures, the nuances of key management become increasingly complex, necessitating a robust strategy that encompasses key rotation, storage, and auditing. Let’s delve a bit more into the overarching concepts of key management in cloud environments, and contrast these with traditional practices and explore the role of key management systems (KMSs) offered by leading cloud providers and solutions.

Key management in cloud environments involves handling various aspects of cryptographic keys, including their creation, distribution, storage, rotation, and deletion. These keys act as digital credentials that facilitate the encryption and decryption of data, ensuring that the information remains unreadable and inaccessible to unauthorized users. Thus, effective key management is crucial for maintaining the integrity and confidentiality of data in cloud systems.

Understanding Key Management Systems (KMS) in the Cloud

Traditionally, key management was often confined within an organization's physical and network perimeters, relying heavily on hardware security modules (HSMs) and manual processes. While this approach was secure, it lacked the flexibility and scalability required in dynamic cloud environments. In contrast, cloud-based key management systems leverage the scalability and accessibility of the cloud, offering integrated solutions that automate many aspects of the key lifecycle management. This modern approach not only enhances security but also adapts to the elastic nature of cloud resources, supporting a wide array of services from data encryption to identity management.

A key management system in a cloud environment serves as a centralized service for managing cryptographic keys, which are essential for data encryption and decryption. KMS solutions are designed to handle the entire lifecycle of these keys, ensuring their secure creation, storage, distribution, and deletion while protecting against unauthorized access. Key features of these systems include streamlined management processes, secure generation and use of keys, and robust access control and auditing capabilities. Such systems are critical components of data security in cloud environments, exemplified by solutions offered by leading cloud providers and specialized solutions like HashiCorp Vault, which provide comprehensive tools to manage and safeguard cryptographic keys effectively.

The operational dynamics of a KMS are critical for maintaining the security and integrity of cryptographic keys throughout their lifecycle. The process begins with the secure creation and storage of keys. When a request is made, the KMS generates a new cryptographic key using secure algorithms, storing it in tamper-resistant storage. This isolation from the data it encrypts is essential because it protects the key from unauthorized access right from its inception.

After a key is securely stored, the KMS serves as a trusted intermediary for performing encryption and decryption tasks. Users or applications can request the KMS to encrypt plaintext data, which the system processes and returns as ciphertext. Likewise, if decryption is needed, the KMS accepts the encrypted data and uses the corresponding cryptographic key to decrypt it, and returns the original plaintext. This design assures that cryptographic keys remain securely within the KMS environment, preventing exposure and maintaining robust security for sensitive data, whether it is in transit or at rest.

One of the KMS's most important roles is key rotation. The system automatically generates new cryptographic keys at regular intervals, replacing the older ones. This routine practice reduces the risk of key compromise over time and helps maintain the integrity of the data encryption process. Furthermore, the KMS implements strict access controls based on the principle of least privilege. This means that only the users and applications you explicitly authorize can do specific things with the cryptographic keys under well-defined rules. In doing so, the system limits potential vulnerabilities and subsequently bolsters the overall security framework.

Another crucial function of any KMS is its full logging and auditing capabilities. All interactions utilizing cryptographic keys, including for their construction, their use, and much more, are meticulously recorded. Enabling this capability provides organizations with audit logs for compliance and security auditing that allows organizations to track who has used their keys and when they have used them to allow you to investigate any potential security incident. KMS provides a robust security infrastructure that meets the highest industry standards and can be audited for compliance with various regulations.

Benefits of Using KMS

The key management system provides a number of benefits that ensure the smooth and secure management of cryptographic keys. One advantage of managing a KMS centrally is having a single interface to manage all your cryptographic keys, which makes the management much easier, as well as drastically minimizing risks for mismanagement or loss. In addition to enhanced security, this centralization also provides support for regulatory compliance tasks via secure state storage, generated key rotation features, and a historical log of all access requests.

These features are essential for both compliance with stringent industry standards and the protection of sensitive data. As an added bonus, the very nature of cloud-based KMS solutions makes them easily scalable, automatically scaling to meet an ever-growing number of keys and instances of encryption without manual intervention.

This scalability makes sure security processes expand alongside organizational requirements. In addition, KMS is cloud-native, which enables greater availability and easy integration with other cloud services, allowing you to easily encrypt and manage your keys across a wide variety of apps and services.

Finally, it abstracts the complexities of cryptographic keys management, providing simple interfaces for encryption and decryption that help minimize the crypto-knowledge needed in development and operational teams. In conclusion, KMS is an essential piece of cloud security protocols, providing a good mix of security, compliance, usability, and scalability.

Best Practices for Cloud-Based Key Management

Combining the operational capabilities of a key management system with secure practices leads to an excellent foundation for secure and compliant cloud-based key management. Establishing strict access controls is key; precise access policies, specifying who is allowed to manage and use cryptographic keys, reduce the risk of exposure and misuse of keys by following the principle of least privilege. KMS also helps automate this process by handling regular key rotation automatically, which periodically refreshes keys to reduce the risk associated with long-term exposure of keys, an important step in maintaining the security of encrypted data.

Moreover, employing KMS for both at-rest and in-transit data encryption protects sensitive data from unauthorized access or interception from the time it is created until it is deleted from the underlying storage systems. Detailed audit logs are also key, providing an immutable record that enables compliance audits and security monitoring, giving the third-party accountability for tracing key usage and changes to encrypted data.

KMS can be integrated with other cloud services to provide scalability and flexibility, which can both simplify encryption throughout the cloud environment and take advantage of the inherent scalability and accessibility of KMS solutions in the cloud. Finally, it is also important to teach staff about cloud security aspects, how to work with cryptographic keys, and what to do when a key is compromised. This key management solution is an all-encompassing method for safeguarding data and in keeping with industry best practices for cloud-based digital asset management.

HashiCorp Vault: A Cloud-Native Key Management Solution

HashiCorp Vault, a key management solution optimized for the cloud-native world with secrets management and data protection capabilities designed to address the challenges of our current dynamic and distributed cloud landscape, illustrates this approach. Unlike other traditional or cloud provider–specific key management systems, Vault is designed especially for the complexities of those specific environments.

HashiCorp Vault provides the most flexibility to organizations for storing and access control enforcement over a wide variety of sensitive data, including, but not limited to, tokens, passwords, certificates, and API keys. This feature works from every cloud you could use to its on-premises hosts; that is the flexibility Vault integrates with.

For example, one of Vault’s remarkable features is dynamic, short-lived secrets. This eliminates the risks that static credentials can carry, which could be compromised over time. Such flexibility is important for upholding high security with speed of operations in environments where it is necessary to maintain a higher degree of automation and velocity.

In addition, encryption itself becomes a much simpler effort through Vault's Encryption as a Service. It allows applications to encrypt data without having to manage cryptographic keys, reducing complexity and increasing data security. Vault abstracts the cryptographic operations so that developers and applications can focus on their core functionalities while it handles the intricacies of data protection. Combining these security measures, HashiCorp Vault is a right fit for organizations looking to secure their cloud-native architectures.

Components and Architecture of HashiCorp Vault

The architecture of HashiCorp Vault and the components that it consists of are built to enable a secure, scalable, extensible approach to secrets management for today's modern cloud-native environments. The core of Vault architecture is Vault Server, which is the primary component that manages an organization's secrets and provides Encryption as a Service. This server serves as the center (hub) to all cryptographic-related operations in the system and ensures that data confidentiality is maintained throughout the system.

The Vault Server is supported by the secrets engines, which are different components for managing different types of secrets (such as AWS credentials or SSL/TLS certificates). Designed to support a range of secret types, a secrets engine provides finely tuned policies to protect and secure secrets based on each secret's sensitivity and purpose. As a result of being modular, Vault can provide tailored security solutions suitable to various operational requirements and technology installations.

The Vault architecture is predominated by different types of authentication methods. These methods validate users and applications for the secrets that are accessed. There are various authentication methods for access control. Vault does not allow the use of sensitive information without verification, so it implements a set of rules, such as tokens, username/password pairs, or cloud IAM roles, among others.

Additionally, HashiCorp Vault excels by providing complete secrets management for multiple cloud providers, as well as on-premises systems. Such flexibility is essential in the current wide landscape of the cloud, where businesses typically run in multicloud environments. The cloud-native design of Vault guarantees that it will integrate perfectly with nearly any infrastructure, helping to secure all areas without disrupting existing workflows.

Moving beyond archaic key management fundamentals and to more contemporary cloud-based solutions is a huge leap in the security of an organization’s digital assets. This not only greatly increases the security but also allows businesses to effectively secure their all-important data and better manage how to spend on the cloud. This strategic approach is crucial for adapting to the fluidity and scale of modern cloud environments, ensuring that security measures keep pace with technological advancements and operational demands.

Network Security Evolution and Segmentation

Network security has come a long way; its basic purpose is to keep data safe while in transit and allow data traffic to happen as required. From traditional physical network segmentation, which relied on hardware appliances to isolate network segments, to virtual LANs (VLANs) and beyond, each era brought forward solutions tailored to the security and operational needs of its time.

Macrosegmentation is the process of splitting the big network into smaller yet broad, distinct security zones based on business needs. Essentially, the main focus of macrosegmentation is to implement and enforce high-level security policies, access control, and traffic flows in between the major segments to reduce the attack surface of networkwide breaches. Typically, this task is achieved by dividing the zones into virtual routing and forwarding (VRFs). On the flip side, macrosegmentation is high-level segmentation, whereas microsegmentation happens at a granular level with segments that can be as small as individual workloads or applications.

The modern approach embraces microsegmentation, enabled by software-defined networking (SDN). This offers granular control over data flow between resources in on-prem environments, traditionally, and now also within cloud environments. It effectively minimizes the attack surface by isolating workloads from each other.

Access control lists (ACLs) are critical components that define which users can access certain resources and where they are allowed to access or not. ACLs have always played a key role in access control in the networks. That said, as networks have grown significantly in size and complexity in recent years, particularly with dynamic cloud environments, managing ACLs has become increasingly difficult. As a result, scalability and manageability have become critical issues that must be dealt with through a more intelligent means of processing ACLs. Though the context is still similar with respect to ACLs, the implementation in current-day cloud-based architectures has changed. ACLs are not just applied at the edge of the network, but deeper in the network—all the way down to the resource or group of resources level.

This transition provides a factor of more local and tenable authority over access, which is important in cloud settings where assets and administrations should be made, changed and expelled continually. Cloud platforms will normally provide this deeper integration using dynamic, declarative approaches. These methods allow rapid configuration changes to the ACLs in response to environmental changes, almost instantaneously. This flexibility is essential for cloud architectures to maintain secure and efficient operations because they determine their response rate toward the introduction of new demands or security threats, which can all together make a large difference in system security and performance. Cloud architectures can provide strict security while still enabling the agility that organizations need to compete in the fast-paced modern technology landscape by enhancing and abstracting the traditional ACL approach.

Cisco’s Application Centric Infrastructure (ACI) revolutionized the way environments in networks are integrated or operated, by joining physical and virtual components under the same policy model. Introduced in 2013 as the first application-centric SDN, ACI pioneered an intent-based policy model that decoupled the implementation of security from the specific network characteristics associated with workloads. This innovative approach has been sustained by alternative SDN solutions, maintaining its relevance in the evolving network architecture landscape.

ACI’s strategy emphasizes modern network segmentation techniques that are agile, scalable, and intuitive, moving away from traditional rigid and manual configurations. This effort is achieved by focusing on the intent of network communications to establish comprehensive access controls. Such a model enables network systems to adapt more dynamically to changing needs and configurations, enhancing both the efficiency and security of network interactions.

Future chapters will explore details of the ACI policy model, its consistency in enabling cloud segmentation, and the interaction of ACI with enterprisewide network management approaches. These deep dive will show how ACI not only meets the current demand but anticipates the networking requirements of the future, ensuring that infrastructure can scale and adapt within the fast-paced, security-conscious environments that define modern IT landscapes.

Infrastructure as Code (IaC) and Security Automation

Infrastructure as Code (IaC) represents a transformative shift in how infrastructure—from networks and virtual machines, compute and storage to load balancers and connection topology—is managed and provisioned, aligning it more closely with software development practices. As a core component of DevOps, IaC allows infrastructure to be treated as if it were software, using machine-readable definition files for management instead of traditional physical hardware configurations or interactive tools. With this method, infrastructure parts can be composed-deployed-managed-versioned just like app source code, which boosts agility and uniformity within operations.

Common tools like Terraform, AWS CloudFormation, and Ansible exemplify this paradigm by enabling teams to automate the provisioning and decommissioning of infrastructure. This automation facilitates rapid deployment and scalability and is critical for embedding security practices directly into the infrastructure from the outset. By integrating security measures into the IaC processes, organizations ensure that these practices are consistently applied across all environments, thereby enhancing overall security posture.

When it comes to applying IaC to security, the primary way is that security policies and configurations are defined as code and allow for their automatic enforcement throughout the infrastructure. In other words, it shifts security from being a secondary consideration, enabling its application at the initial stages of deploying infrastructure. In that regard, crucial advanced tools include HashiCorp's Sentinel, and the Open Policy Agent (OPA), as they make it possible to adopt the policy-as-code approach that allows integrating security into the IaC's central lifecycle.

The OWASP Infrastructure as Code Security Cheat Sheet offers an extensive array of best practices for enhancing the security of IaC deployments. It covers essential topics such as version control, automated testing, the principle of least privilege, and comprehensive auditing. These guidelines are designed to ensure that the deployment of infrastructure as code not only speeds up the process but also integrates robust security measures from the start. For a detailed exploration of these practices, see the "Infrastructure as Code Security Cheatsheet" (refer to the "References" section at the end of the chapter.)

Advanced Load Balancing and Application Layer Security

We will review advanced load balancing and application layer security using AWS's Load Balancer Services solutions as a reference, because it is an extremely mature and massively adopted solution in the market.

Cloud providers differentiate between two types of load balancers. For HTTP/HTTPS traffic, you have application load balancers (ALBs), and for raw TCP/UDP traffic, you can use network load balancers (NLBs). In fact, if you require mind-numbingly fast performance, NLBs are where the magic happens, whereas ALBs are ideal for web apps! NLBs can easily withstand millions of requests per second without flinching, and they're awesome for those spikes of random traffic. They have some core capabilities that make them a tempting choice: TLS termination, preserved client IPs, and exciting stability across zones. NLBs can handle long-lived TCP connections or WebSocket apps. An NLB acts like a clever network router rather than a standard load balancer. Rather than audaciously messing with an HTTP request, it works its magic at the packet level. When clients hit through an NLB, they deal with your server somehow.

AWS's application load balancers significantly enhance application scalability and reliability by efficiently distributing incoming traffic across various targets in multiple availability zones, operating at the application layer of the OSI model. These advanced load balancers optimize routing mechanisms through content-based routing decisions that include HTTP headers, methods, and paths, while also boosting application performance and reliability with intelligent application-level health checks and support for dynamic services like serverless functions and gRPC over HTTP/2 transport. ALBs seamlessly integrate with cloud-native services and architectures, incorporating critical security features such as SSL/TLS decryption to offload these tasks from application servers, thereby centralizing the encryption and decryption processes. This integration strengthens application security by providing robust protection against application-layer attacks, including cross-site scripting (XSS) and SQL injection.

When you're choosing between an ALB and an API gateway, the decision hinges on the specific requirements of the application. ALBs are particularly well-suited for managing HTTP and HTTPS traffic within scalable, high-performance web applications, offering enhanced control over traffic distribution. In contrast, API gateways provide more granular control over API traffic, featuring API version control, key and rate limiting, and direct integration with AWS Lambda for serverless deployments. The use of ALBs in cloud architectures typically involves distributing incoming web traffic across multiple backend services to optimize performance and reliability, ensuring efficient traffic management and enhancing security measures through SSL/TLS termination and advanced routing rules that safeguard against various types of attacks.

Application Load Balancers (ALBs): Features and Use Cases

The application load balancer is a fundamental service in AWS, which provides intelligent routing to applications on Layer 7 of the OSI layer. With scalable performance required to execute complex routing, security, and monitoring use cases effectively, ALBs empower organizations. Besides their basic functionalities, ALBs can be the cornerstone for modern architectures such as microservices, containerized environment, and serverless applications. Some key features include

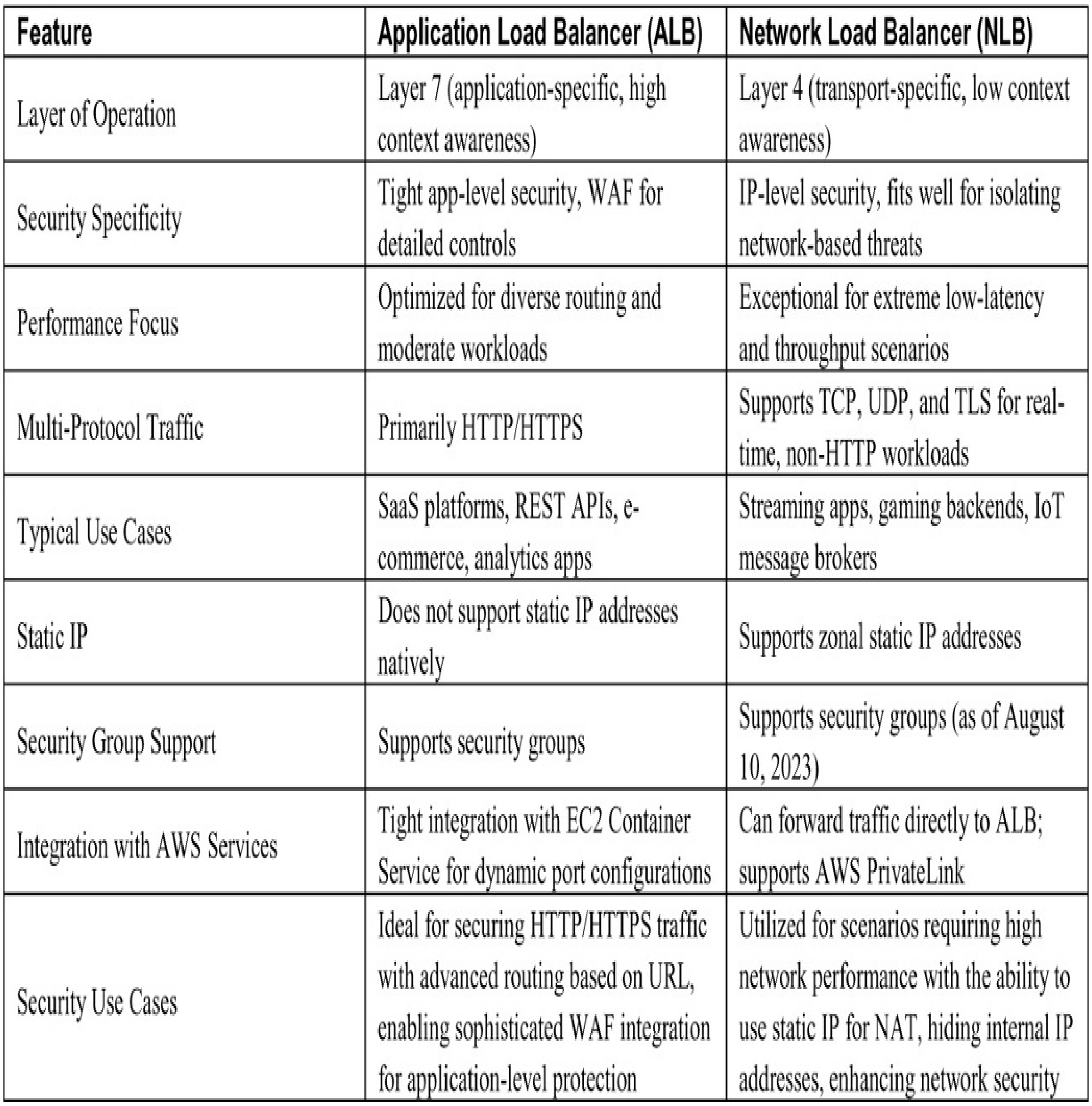
1. **Fine-Grained Routing Decisions:** ALBs allow for advanced, content-based routing in that they can route based on HTTP headers, query parameters, and even method types. This enables you to redirect the traffic intelligently between microservices or backend as per user requirements or geo data.
2. **Host and Path-Based Multitenancy:** By leveraging host and path-based routing, you can route requests to multiple different applications all behind a single ALB instance. This capability is particularly useful for multitenant environments or where you are running multiple versions of your application (staging and production) and want to segment their workloads in a very cost-effective and operationally efficient manner.
3. **Ability to Handle Protocol in Real Time:** ALBs provide the native infrastructure for latency-sensitive applications, such as chat systems, live dashboards, or publishing services, by supporting HTTP/2 and WebSocket protocols.
4. **AWS Deployment Hooks:** ALBs are designed to work seamlessly with AWS-native services. Whether it is Elastic Kubernetes Service (EKS) or Elastic Container Service (ECS), the ALB dynamically registers/deregisters target groups based on the state of a task or pod.
5. **Support for Serverless and Lambda:** ALBs extend their use case to serverless applications by offering AWS Lambda as a target. This allows organizations to route requests to compute resources without provisioning traditional backend servers.

6. **Security at Scale:** Capabilities such as SSL/TLS termination make it easier to manage certificates while adding security at the application layer. When used with AWS Web Application Firewall (WAF), ALB stands against OWASP Top 10 dangers, such as cross-site scripting and SQL injection.
7. **Complete Observability:** ALBs integrate with Amazon CloudWatch for detailed request metrics and traffic pattern metrics in a granular manner. Logs and metrics are essential for diagnosing problems in real time and minimizing resource use.

The application load balancer is widely applicable for many modern application usage scenarios. For a microservices use case, the ALB is the solution that can assist in load-balancing traffic between registered services along with its content-based routing features, which can tell requests how to reach its respective microservice endpoints based on path or header parameters. For web application API gateway use cases, the ALB acts as the unified entry point for APIs, offering advanced routing and pinch-free SSL/TLS termination. The ALB also integrates into the architecture of multiregion failover; when it is used in conjunction with Route 53, high availability is achieved by routing end users to the nearest healthy region in the event of a regional outage. ALB's dynamic target group update capabilities are a natural fit for containerized environments, with seamless scaling up and down of health and availability of ECS tasks or EKS pods. In addition, real-time applications like collaborative tools or live games leverage ALB's WebSocket protocol, which supports stable connections crucial to users' experience.

Table 15-2 can help you refine your understanding of where each type of load balancer excels, particularly from a security standpoint.

Table 15.2 *ALB vs. NLB with Security Use-Cases*



Feature	Application Load Balancer (ALB)	Network Load Balancer (NLB)
Layer of Operation	Layer 7 (application-specific, high context awareness)	Layer 4 (transport-specific, low context awareness)
Security Specificity	Tight app-level security, WAF for detailed controls	IP-level security, fits well for isolating network-based threats
Performance Focus	Optimized for diverse routing and moderate workloads	Exceptional for extreme low-latency and throughput scenarios
Multi-Protocol Traffic	Primarily HTTP/HTTPS	Supports TCP, UDP, and TLS for real-time, non-HTTP workloads
Typical Use Cases	SaaS platforms, REST APIs, e-commerce, analytics apps	Streaming apps, gaming backends, IoT message brokers
Static IP	Does not support static IP addresses natively	Supports zonal static IP addresses
Security Group Support	Supports security groups	Supports security groups (as of August 10, 2023)
Integration with AWS Services	Tight integration with EC2 Container Service for dynamic port configurations	Can forward traffic directly to ALB; supports AWS PrivateLink
Security Use Cases	Ideal for securing HTTP/HTTPS traffic with advanced routing based on URL, enabling sophisticated WAF integration for application-level protection	Utilized for scenarios requiring high network performance with the ability to use static IP for NAT, hiding internal IP addresses, enhancing network security

While ALB and NLB are both critical in distributing traffic, they are meant for different use cases and operate at different layers of the OSI model. As we have mentioned, NLB operates at Layer 4, which is great for ultra-low latency and high-throughput workloads. This is great for applications that depend on TCP or UDP traffic, especially those requiring features such as static IP addresses or PrivateLink integration. By contrast, ALB's operation at Layer 7 makes it appropriate for new application architectures that call for advanced routing features to application-layer protocols, such as HTTP/HTTPS, as well as deep content-based functionality.

The ALB excels in situations where application-layer intelligence is required. It also enables content-aware routing based on user identity, session metadata, or type of content for more granular control over traffic and user experiences. On the other hand, NLB is the one you should use for ultimate applications that need more raw performance, less latency, and non-HTTP protocol support.

In most cases, the best results are achieved by deploying a mix of ALBs and NLBs. As an illustration, an ALB can intelligently route a user-facing request among several services depending on application logic, and an NLB might serve high-throughput or low-latency traffic to a service (think streaming platform or IoT data pipeline). This multilayered architecture enables the best performance and scale for different classes of workloads.

The Future Landscape: Why OSS ALBs and Ingress Controllers Are Gaining Traction

As the world increasingly adopts Kubernetes (K8s) for orchestrating containerized applications, the role of open-source application load balancers and ingress controllers indeed becomes more critical and arguably more dominant in the cloud-native ecosystem. Much of this transition is happening due to the fundamental demands of Kubernetes deployments for effective traffic control, security, and scale—needs that many of these tools are particularly well-suited to meet.

As the modernization of cloud-native architecture is happening, open-source software (OSS) ALBs and ingress controllers are gradually becoming the new norm due to their deeper integration with Kubernetes. As a result, Kubernetes lends itself to a microservices architecture that breaks up applications into smaller, independently deployable services. OSS ALBs and ingress controllers assist in achieving this by providing seamless traffic routing, granular load balancing, and enhanced observability, which are essential for ensuring high performance and reliability in microservices-based applications. Their ability to adapt to dynamic scaling and integrate with modern CI/CD pipelines makes them indispensable tools in the cloud-native ecosystem.

Additionally, these open-source offerings provide maximum flexibility and customization to tailor the load-balancing capabilities to the specific needs of your Kubernetes environment. The customization of OSS ALBs and ingress controllers allows you to adapt them to suit different needs and states. Whether it is specific routing requirements, SSL/TLS termination, or traffic management, deployments are always optimized for performance, security, and reliability. The large user community around Go-based OSS ALBs and ingress controllers like Traefik, NGINX, and HAProxy helps to drive improvement and innovation. These communities are there to keep these tools on the edge of tech with other very important features, security best practices, and so on, to Kubernetes applications.

The cost-effectiveness is another major advantage of open-source tools. They provide significant cost advantages, making them appealing to both startups and enterprises looking to expand Kubernetes deployments with minimal financial impact compared to proprietary or cloud provider–specific options. Finally, if you deploy and manage these tools on any infrastructure, you run the risk of preventing vendor lock-in, and we can argue that makes those tools economically sound.

Security stands as a vital consideration in cloud-native frameworks, and OSS ALBs and ingress controllers enhance security by offering the robust security features integral to Kubernetes deployments. Known features include SSL/TLS termination and integration with K8s Secrets to manage certificates securely and support WAFs to secure against web vulnerability. The openness of these tools also enables the community to quickly discover and fix security vulnerabilities.

Finally, OSS ALBs and ingress controllers are built to integrate with components in the broader Kubernetes ecosystem, including CI/CD, monitoring, and service meshes. OSS ALBs and ingress controllers are compatible with many other tools, increasing the agility and efficiency of DevOps. These integrations are important for running and maintaining a unified and effective operational culture in cloud-native deployments.

The landscape of open-source load balancers is diverse and well-suited to a broad spectrum of applications, ranging from traditional web deployments to sophisticated, containerized microservices architectures. Such richness provides organizations with a range of options tailored to their unique requirements. Let’s take a look at a few notable open-source load balancers:

- **Traefik:** Traefik is renowned as a “cloud-native edge router,” making it an excellent choice for microservice-based environments. It integrates seamlessly with container orchestration platforms like Kubernetes and Docker Swarm, offering dynamic configuration and a rich set of middleware options. These features are essential for modern application infrastructures that demand flexibility and rapid scalability.

- **NGINX:** NGINX is a versatile tool that functions as a web server, reverse proxy, and load balancer. It is widely favored for its lightweight design and remarkable scalability, efficiently handling high levels of HTTP/HTTPS traffic. NGINX’s modular architecture and extensive community support contribute to its adaptability, making it a popular choice in both local and cloud-based microservices environments.

- **HAProxy:** HAProxy is particularly effective in high-traffic scenarios, leading the industry in balancing TCP- and HTTP-based applications. Known for its reliability and precision, HAProxy provides comprehensive metrics and logging, empowering organizations to optimize performance and maintain high availability. Its robust feature set makes it a go-to solution for mission-critical applications.

- **Seesaw:** Developed by Google, Seesaw is designed for internal traffic management in large-scale systems. It is inspired by enterprise-grade engineering practices and offers clear, straightforward methods for network load balancing. While its features are somewhat specialized, Seesaw provides a powerful option for complex enterprise networking needs.

- **Neutrino:** Originating from eBay’s operational requirements, Neutrino is less well-known than other load balancers but provides powerful functionalities through its JVM runtime. This design ensures cross-platform compatibility, making it an attractive choice for organizations seeking flexible and scalable load balancing solutions.

Together, these open-source solutions improve network management by providing automotive tools that scale, are flexible, and fit the ever-changing demands of contemporary digital infrastructures. These tools enable foundational functionality for all network operations, whether for small startups or enterprise networks, delivering strong security and agility in networking services.

Kubernetes's adoption of OSS ALBs and ingress controllers is sure to only increase as Kubernetes establishes itself as the de facto platform for deploying and managing containerized applications. As modern, dynamic applications are being deployed on Kubernetes, it makes sense to allow sophisticated load-balancing, traffic management, and security features to be offered in a flexible, cost-effective manner. In addition, the innovation and community support around these open-source workhorse tools make them quite robust in tackling the ever-changing odds of cloud-native deployments. The decision to go with either ALBs or NLBs really depends on the needs of your application and your infrastructure team—whether you need advanced control over traffic routing, minimal latency and jitter, or support for transport-layer protocols. At the same time, the extensible open-source load-balancer ecosystem provides a set of flexible and powerful solutions to real load-balancing problems for those who want to take load balancing in a direction beyond what a cloud provider can deliver.

The Cloud-Native Security Stack: From Infrastructure to Application

Creating a secure cloud-native environment requires a holistic approach—one that encompasses much more from the ground level right up to the apps running on top, as well as the end users accessing those apps. Given an IT landscape that is defined today by a mix of traditional, containerized, and serverless architectures, this broad security posture is required. Let's now take a high-level view of security in the cloud-native stack, examine what each layer means, consider possible ways to secure, and investigate Cisco's solutions as examples.

1. **Cloud Infrastructure Security:** Securing cloud infrastructure is fundamental, covering protection at the network, storage, and

compute levels. It includes protecting things such as the virtual private cloud (VPC) as well as implementing strong access controls and extensive data encryption in transit and at rest. Solutions that work well for this include next-generation firewalls (NGFWs) that offer strong perimeter protection and cloud security posture management (CSPM) tools that can continuously monitor and enhance your security posture. For instance, Cisco Secure Firewall and Cisco Secure Cloud Insights deliver critical north-south protection and CSPM capabilities, respectively, for these foundational security measures.

2. **Container and Kubernetes Security:** As containerization and Kubernetes become more prevalent, security measures must extend beyond traditional perimeter defenses to include container runtime protection and image scanning. These practices are essential for detecting vulnerabilities before deployment and preventing malicious behaviors during runtime. Additionally, securing Kubernetes environments requires robust cluster configurations, enforcing network policies, and implementing strict access controls. To address these needs, dedicated container security solutions and specialized Kubernetes security tools are crucial for maintaining secure deployments and enforcing network segmentation. Cisco Secure Workload enhances security through microsegmentation within clusters, effectively isolating workloads to minimize attack surfaces. Furthermore, Cisco Secure Firewall Cloud Native (SFCN) provides targeted security for Kubernetes environments, offering granular control over east-west traffic within the cluster. Together, these solutions deliver comprehensive protection for containerized applications and Kubernetes deployments, ensuring resilience against evolving threats.

3. **Serverless Security:** Serverless architectures offer scalability and efficiency but introduce unique security challenges such as function permissions, dependencies, and API vulnerabilities. The transient nature of serverless functions requires innovative security approaches focused on real-time monitoring and protection. Examples of solutions in this space include serverless-specific security platforms that deliver function-level security, automated

vulnerability scanning, and API security. One of these is Cisco Secure Application, which works with serverless platforms such as AWS Lambda to provide real-time visibility and security vulnerability management functionality.

4. **Application Security:** Application security is important across the entire software development lifecycle, starting as early as code development and going all the way through to deployment. This one is all about securing your application code, securing the dependencies to make sure they are not causing any harm. It is also used to protect APIs and the communication between microservices. Tools provide solutions for static and dynamic application security testing (SAST/DAST), runtime application self-protection (RASP), and API security, which are crucial. As another example, Cisco's Secure Application for AppD provides runtime security and dependency management, and still other products like CNAPP solutions serve the role of an API security solution.

5. **Identity and Access Management (IAM):** IAM helps you set up rules to allow or disallow access to different resources within the cloud-native environment. This includes controlling identity management, applying granular access policies, and applying multifactor authentication (MFA) and single sign-on (SSO) features. Comprehensive IAM platforms catering to cloud-native architectures are essential, providing fine-grained access controls, MFA, and SSO. An example of this is Cisco Secure Access by Duo, which offers strong MFA and secure access visibility, to provide trust for users and devices.

6. **Data Security and Compliance:** As the name suggests, this layer is responsible for the protection of sensitive data across cloud-native environments and for compliance with relevant regulations and standards. This involves data encryption, secrets management, and compliance monitoring platforms. Cloud data security tools such as Cisco Secure Cloud Insights can assist users in managing, auditing, and reporting compliance across multiple environments to fulfill stringent regulatory data security requirements.

7. **End-User Security:** Securing the end-user experience means protecting against bad actors performing phishing, managing secure access to applications, and monitoring for anomalous behavior. Key solutions within this application security segment include web gateways, user behavior analytics, and secure application access. Cisco Duo Network Gateway and Cisco Secure Access by Duo ensure granular control of user and endpoint access in a way that can establish strong trust without traditional VPNs.

Securing the cloud-native stack from infrastructure to application is a layered exercise, because the unique challenges and vulnerabilities at each level dictate this design philosophy. By everaging end-to-end security offerings across infrastructure, container, serverless security, application security, identity security, data protection, and end-user security, organizations can develop a comprehensive security posture to protect their cloud-native environments against evolving threats. Just as an example of how tools that were developed to protect the entire stack of a cloud-native solutions can work well together, Cisco's suite of security solutions, integrated at the infrastructure level, provides a blueprint of a secure cloud-native ecosystem.

We will discuss cloud native application security in depth in Chapter 16. This discussion serves as an introduction to the potential threat vectors in a multilayered approach to securing native cloud resources and application development.

Navigating Multicloud and Hybrid Cloud Security

The challenges of multicloud and hybrid cloud demand their own set of services, employees, and advanced technologies that need to be leveraged to secure such infrastructure. Organizations are increasingly challenged with the management and security of these environments, as IaaS and multicloud strategies gain speed.

Employing multicloud security tools and best practices ensures consistent security policies and controls across different cloud providers, but in a hybrid cloud environment, additional considerations will apply, such as how to best balance and optimize the security implementation across the different toolchains and network locations, leveraging the strengths of both on-premises and cloud-based solutions.

Effective cloud security is dependent on the implementation of well-defined policies that govern cloud ownership, risk acceptance, and responsibility across multiple clouds. These policies need to be platform specific and address each unique security control and configuration required within those environments. Getting consistency of security posture across multiple cloud providers is important too, since the different security configurations may expose the assets to various risks. Choosing the right security tools is paramount—mainly those that sync security policies across platforms and aid in compliance by offering end-to-end visibility and control over security posture across all environments.

Additionally, it is an important strategy to automate routine security tasks to strengthen cloud security, particularly considering that human error—a significant contributor to numerous cloud security breaches—is often implicated in security incidents. It not only makes the entire operation efficient but also helps provide a seamless enforcement of security measures. In a hybrid cloud, organizational IT comprises a mix of the on-premises infrastructure along with the public cloud service. Deploying security in hybrid cloud environments requires a nuanced approach. First, organizations need to unify their security posture across on-premises and cloud elements. Cisco Secure Firewall and Cisco Secure Workload can secure these environments by providing network security while facilitating microsegmentation and policy enforcement.

Solutions that provide a complete range of capabilities supporting all areas of cloud security, such as Cisco Secure Application for application security and Cisco Secure Access by Duo for identity and access management, add tremendous protection to cloud environments. These tools were created to cover various aspects of cloud security to help ensure organizations move confidently into multicloud environments using cloud security tools that help them maximize what the cloud has to offer with minimal risk. Such a unified methodology provides protection from the infrastructure level all the way to the application level across multiple cloud and hybrid environments.

Advanced Security Measures and Third-Party Services

Multicloud security is all about being able to control the nuances of types of isolation and security in every particular cloud you use: the first step thus being multicloud native security. Due to misconfigurations and user error, Gartner has forecast that a staggering 99 percent of cloud security failures will be blamed on customers (see “Is the Cloud Secure?”; refer to the “References” section for more details). Additionally, the growing trend to use multiple cloud providers as part of a multicloud strategy makes competitive comparison of static cloud service offerings less valuable for sustaining similar postures.

That makes consistency of policies, a thorough understanding of the strengths of each platform in terms of security, and a strict application of security measures in any cloud service absolutely essential. Such intricacy, hence, explicitly highlights the need for better security proposals that are not restricted to just identification and resolution of risks but formulate a wider security perspective, covering all on-premises and even cloud atmospheres on a tactical basis.

Such complications call for an end-to-end solution such as that provided by cloud security posture management (CSPM) and cloud workload protection platforms (CWPP), both of which facilitate unified security management across cloud and on-premises environments.

The Need for Cloud Security Posture Management (CSPM)

As organizations adopt cloud environments for their scalability, flexibility, and economy, they also encounter unique security challenges that traditional tools can barely tackle. Since resources are provisioned, adjusted, and retired without delay in a cloud environment, efficiently maintaining manual scrutiny over its configuration is impractical given how dynamic (and complex) cloud infrastructure can be. Misconfigurations, like exposing an S3 bucket or misconfigured IAM roles, continue to be among the top causes of cloud breaches, and sensitive data is at risk. To overcome these challenges, cloud security posture management has become an important tool for the first task, combining strong fundamental capabilities with transformative benefits for securing next-generation cloud environments.

Features of CSPM

CSPM solutions improve high visibility in cloud security and help secure the cloud environment. These solutions include configuration management that quickly identifies and remediates misconfiguration and also the integration of up-to-date threat intelligence to address new threats. Policy enforcement validates compliance with regulatory requirements such as GDPR and PCI DSS as well as the organization's policies. CSPM tools are built on the principles of scalability and can meet the complexity and scale of modern cloud environments, giving organizations a single centralized platform for managing multiple cloud providers. They are integrated into DevOps workflows to weave security into the fabric of the development lifecycle via secure Infrastructure as Code. Built-in reporting and analytics improve decisions and accelerate audit processes and maintains security posture.

Benefits of CSPM

Unlike typical security solutions, CSPM provides real-time assessment and continuous monitoring to ensure rapid discovery of misconfigurations and vulnerabilities in cloud environments. CSPM provides unified visibility of cloud resources to eradicate blind spots, and it enables you to address security gaps faster. Automated remediation reduces manual effort by enabling you to respond to security threats—such as excessive permissions or unpatched boxes—quickly and consistently. This simplifies compliance management by continuously assessing against regulatory guidelines and generating full compliance reports. With risk prioritization capabilities, teams can get actionable, risk-based context about how serious and impactful security issues are, allowing them to focus on remediating critical threats first. Additionally, CSPM maintains security management across multiple platforms that constantly ensure efficiency and practicality for an organization that is functioning on different cloud technologies.

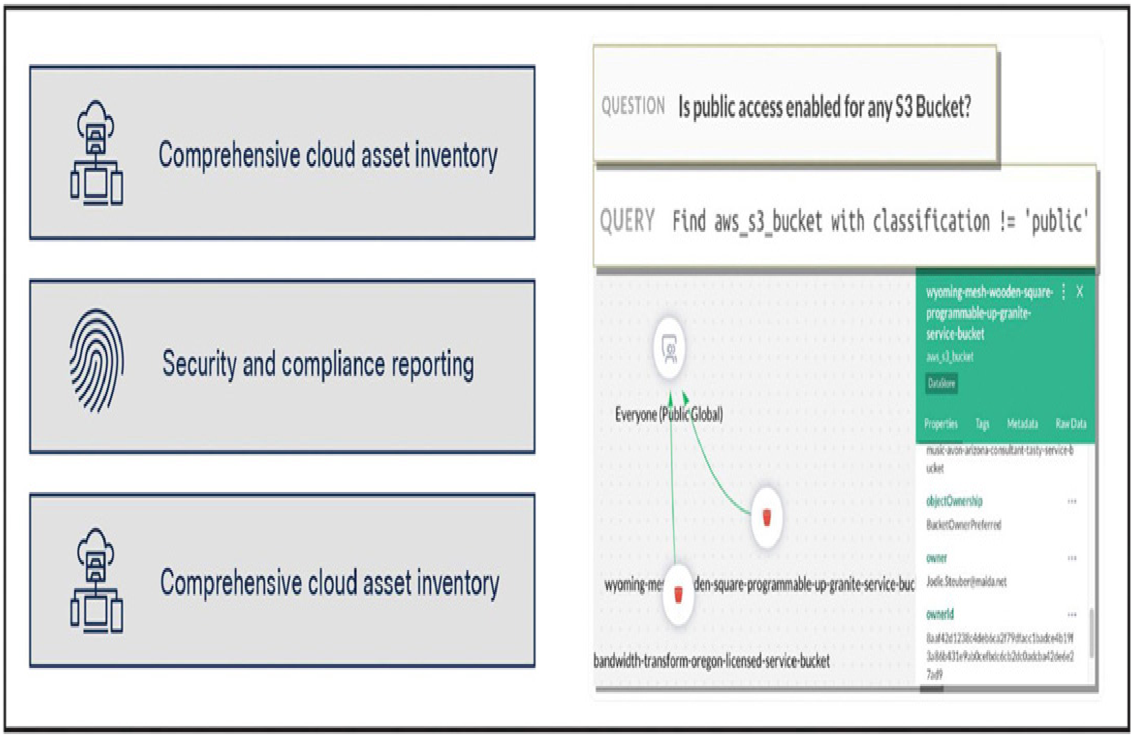
The Future of CSPM

As more businesses migrate to the cloud, CSPM is only going to become more important in maintaining the security and compliance of their environments. Artificial intelligence and machine learning have been themes at the forefront, and the CSPM compass will evolve to encompass predictive analytics along with action items that can be automated to assist in proactively eliminating the threats. CSPM mitigates developing threats while advancing inside the cloud, making it an indispensable future-proofing tool for contemporary cloud security techniques

Integrating Cisco Solutions: Enhancing Multicloud and Hybrid Cloud Security with Attack Surface Management and JupiterOne

Cisco has recognized the importance of effective cloud security posture management in addressing the complexities of securing hybrid cloud environments. For this reason, Cisco created its own CSPM solution by virtue of its joint product partnership with JupiterOne, a cyber asset management and governance platform that enables organizations to gain unified visibility, relationship mapping, and security insights across the digital ecosystem. Out of this collaboration, Cisco Attack Surface Management (Cisco ASM) was born—a solution that gives organizations the ability to gain full visibility, automate detection of compliance violations, and enable preemptive detection of incoming threats. With Cisco ASM built in, organizations also get a dynamic and detailed cyber asset inventory to gain visibility, management, and enforcement for their multicloud and hybrid cloud strategies.

Figure 15-3 demonstrates how a “simple” conceptual question (“Is public access for any S3 Bucket?”) can be answered almost instantaneously using natural language queries (NLQs). A natural language query that is input consisting solely of terms or phrases spoken normally or entered as they might be spoken, which drastically simplifies operations.


Comprehensive cloud asset inventory
Security and compliance reporting
Comprehensive cloud asset inventory

QUESTION Is public access enabled for any S3 Bucket?

QUERY `Find aws_s3_bucket with classification != 'public'`

Everyone (Public Global)
wyoming-me den-square-programmable-up-granite-service-buc
bandwidth-transform-oregon-licensed-service-bucket
wyoming-mesh-wooden-square-programmable-up-granite-service-bucket
aws_s3_bucket
DataStore
Properties
Tags
Metadata
Raw Data
music-avon-arizona-consultant-tasty-service-bucket
objectOwnership
BucketOwnerPreferred
owner
Jodie.Steuber@maida.net
ownerId

Figure 15-3 *Introducing Cisco Attack Surface Management—Querying Relationship Mapping of Cloud-Based Entities and Access Rights*

Cisco ASM employs machine learning and analytics-powered technology that runs continuously, scanning through cloud environments. It enables the detection of misconfigurations, compliance violations, and possible security risks, thus improving the security posture of organizations. This feature helps get a better understanding of the organization's attack surface position. Cisco ASM provides a complete inventory of assets, providing deep visibility across multiple environments into all cloud resources and where they are deployed. It also automates compliance as code monitoring, capable of enforcing cloud deployments to remain in compliance with multiple regulatory standards and security controls.

Complementing this capability, Cisco ASM senses misconfigured workloads and drives speedy remediation with actionable insights to alleviate security risks. It is especially designed for hybrid environments, giving the integrated option for both cloud and on-premises infrastructures providing centralized security management. Advanced algorithms that drive enhanced threat detection features identify suspicious activity and potential threats, allowing security teams to respond quickly and effectively.

Naturally, the Cisco ASM is designed to be part of the broader Cisco Secure portfolio and can operate seamlessly with Cisco XDR (all key capabilities will work with Cisco SecureX) and can leverage Device Insights capabilities built in at the Cisco XDR layer. This powerful cohesion allows easier visibility and compliance management; plus, it helps with threat detection before attack. With tools as advanced as these, organizations can feel more confident in their ability to meet the complexities of cloud security and to ultimately ensure that their cloud deployment is secure, compliant, and positioned for agility when the unknown comes calling.

Cloud Workload Protection Platforms (CWPPs) and Cisco Secure Workload

Cloud workload protection platform solutions are specifically designed to protect workloads in the cloud (physical and virtual machines, containers, serverless functions). In contrast to CSPM, which focuses on the external security posture and compliance of the cloud environment, CWPP focuses solely on the internal security of the workloads. It also includes runtime protection, system trust, and network segmentation.

Cisco Secure Workload (formerly Cisco Tetration) is an application workload security solution for hybrid multicloud workloads, providing visibility and dynamic protection. It is designed specifically for three key use cases:

1. The first on the list is zero trust microsegmentation, which is a fundamental capability of Secure Workload. It uses both agent and agentless methods for workload discovery through labels and offers segmentation policy recommendations based on traffic flows. This enables policies to be validated and tested without affecting operations and enforces these dynamic policies across multiple enforcement points. They consist of elements like host-based firewalls, data processing units (DPUs), network firewalls, load balancers, and built-in security controls of the cloud to create a wide security perimeter.

2. Second, the Secure Workload Vulnerability Detection and Protection capability uses an agent to observe the application workload at

runtime. This allows for detection of packages and container images that have vulnerabilities. This information is used to enforce vulnerability (CVEs) attribute-based policies that can either quarantine workloads or can even perform virtual patching with Cisco Secure Firewall. Not only does this approach identify risks, but it goes further to mitigate them, which improves the security framework.

3. Finally, Behavioral Detection and Protection observes running processes for behavior changes and generates a behavior process tree and behavior snapshot. It identifies suspicious behavior based on the mitigation tactics, techniques, and procedures (TTPs) defined by MITRE ATT&CK or a custom forensic rule set. Integrated with Rapid Threat Containment of Secure Firewall, it protects agent and agentless workloads from emerging threats based on behavioral anomalies so that the operational environment is safe from sophisticated and evolving threats. (See the “Cisco Secure Workload and Secure Firewall White Paper”; refer to the “References” section for more details.)

A solution overview, as illustrated in Figure 15-4, highlights the extensive coverage and consistency achievable across any infrastructure or location. For a deeper dive into the specifics, please refer to the “References” section at the end of the chapter.

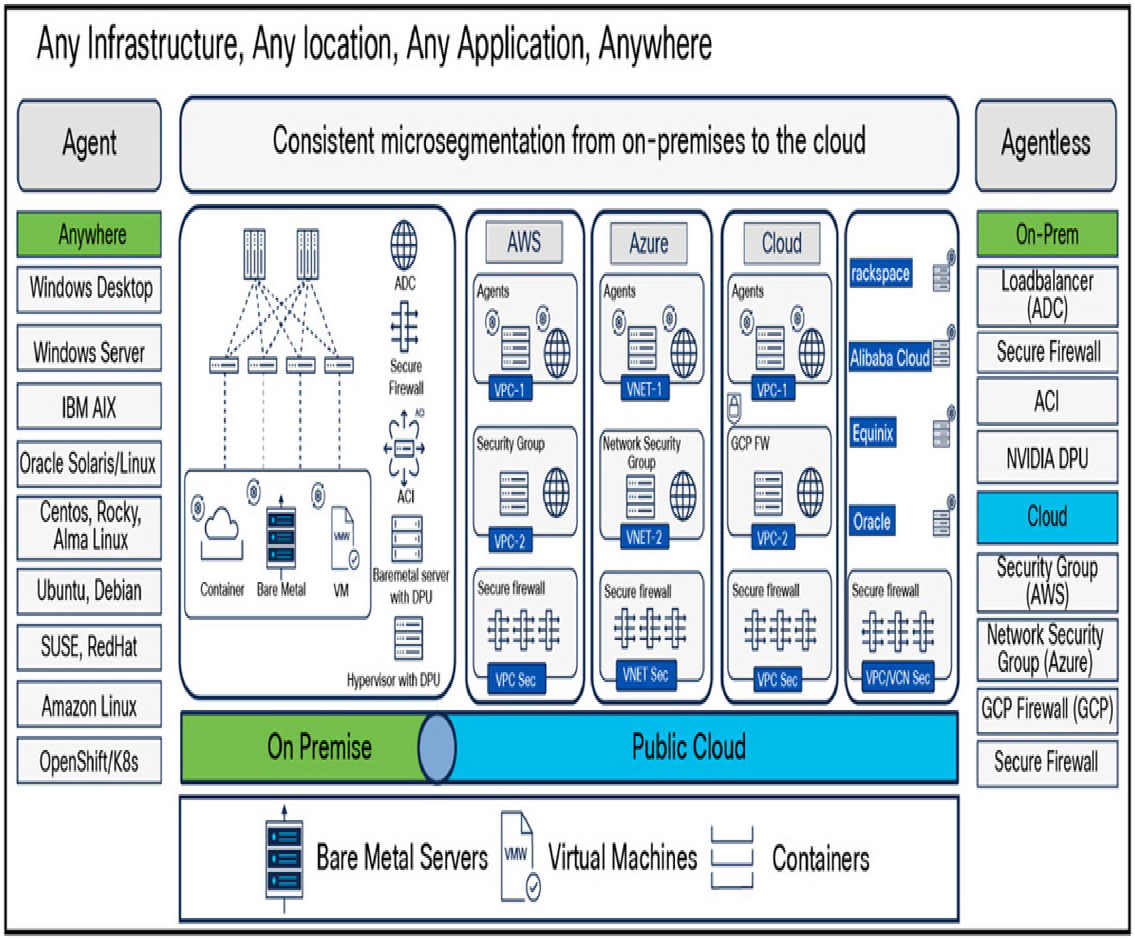


Figure 15-4 *Cisco Secure Workload—Consistent Microsegmentation Across Environments*

Cisco Secure Workload embodies the CWPP philosophy by providing granular visibility and control over all workloads, regardless of their location in hybrid environments. This visibility is foundational for implementing zero trust security models, because it allows for precise segmentation, continuous monitoring, and enforcement of least-privilege access controls at the workload level.

Cisco Secure Workload features a comprehensive set of capabilities focused on securing your network. Its main feature is microsegmentation, which analyzes app dependencies and traffic flows to identify precise policies that prevent workloads from talking to each other, thereby greatly reducing lateral threat movement across the network. Finally, Cisco Secure Workload also uses machine learning to automatically generate and enforce security policies so that workloads have only the permissions and access they need, consistent with the zero trust security model. These capabilities are further augmented with compliance and risk management functions that assist with continuous monitoring and reporting for regulatory compliance and compliance with internal policies, which is an impactful internal security component that many CSPM solutions do not address.

Relationship to Zero Trust

Moving to zero trust security models is making cloud-native security paradigms undergo foundational changes. The idea behind zero trust is that no trust is granted automatically, whether the resources are inside the corporate network or not. Together, two solutions form this approach: CWPP protects the internal cloud workload, and CSPM protects the external cloud environment and posture.

Cisco Secure Workload's capabilities, from microsegmentation to policy automation, are directly aligned with zero trust principles. Cisco Secure Workload delivers comprehensive visibility into workload behavior and fine-grained control over inter-workload communications to ensure that only trusted, authorized, and necessary communication is allowed in the cloud environment.

Going Up the Stack from Infrastructure to Application

Figure 15-5 showcases how Cisco's security products integrate within an organization's overarching security strategy, providing a layered and comprehensive approach to safeguard cloud-native applications in diverse environments, including multicloud and hybrid setups. This multilayered strategy deploys a range of security technologies, described next. These technologies contribute to Cisco Secure's portfolio, offering enhanced visibility and enforcing stringent security policies tailored for cloud-native ecosystems.

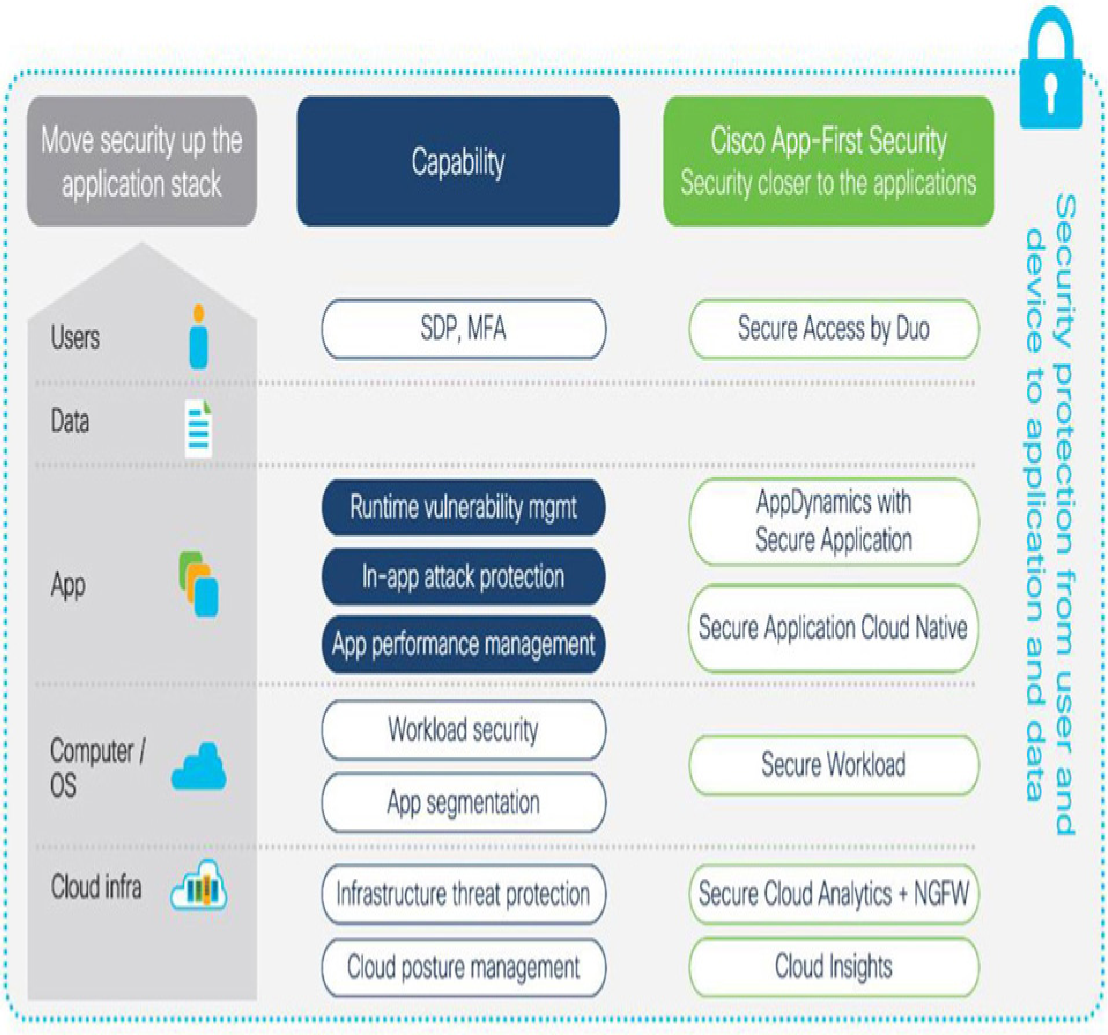

Move security up the application stack
Capability
Cisco App-First Security
Security closer to the applications
Users
Data
App
Computer / OS
Cloud infra
SDP, MFA
Secure Access by Duo
Runtime vulnerability mgmt
In-app attack protection
App performance management
AppDynamics with Secure Application
Secure Application Cloud Native
Workload security
App segmentation
Secure Workload
Infrastructure threat protection
Cloud posture management
Secure Cloud Analytics + NGFW
Cloud Insights
Security protection from user and device to application and data

Figure 15-5 *Secure Cloud-Native Infrastructure Using Cisco Secure Portfolio*

- **Cisco Secure Cloud Native Application Protection (CNAPP):** In the cloud-native ecosystem, securing applications is critical. Cisco Secure Application is adeptly integrated into the development pipeline, delivering comprehensive security analytics and CI/CD pipeline integration, along with enhanced Kubernetes and container security. This includes API visibility and risk detection capabilities embedded directly within the application runtime environment. It plays an essential role in real-time vulnerability identification, dependency management, and the prevention of exploits. This ensures ongoing protection and security for applications operating within the diverse landscapes of multicloud and hybrid environments.

- **Cisco Cloud Application Security:** This platform is the latest evolution of Cisco's CNAPP; it unifies security through a platform that merges CSPM, CWPP, API security, and IaC security. It addresses the need for comprehensive coverage from code to cloud, delivering protection throughout the entire application lifecycle and across all cloud assets. This platform emphasizes contextual risk prioritization and dynamic remediation guidance, enhancing the capacity to effectively identify and mitigate risks. With this evolution, Cisco Cloud Application Security provides an all-in-one security solution that aims to reduce risks and costs while improving productivity by offering a consolidated view of security postures and simplifying threat management across cloud-native architectures.

- **Application Security with Cisco Secure Application (AppD):** With Cisco Secure Application for AppDynamics, the application security focus sharpens on the continuous detection of code dependencies and the monitoring of vulnerabilities. As a part of the AppDynamics suite, it operates within the Application Performance Monitor (APM) and is deployed as an agent inside the application code itself. Such an integration enables constant security monitoring in the application runtime, allowing the solution to block exploits in real time as they are recognized. Because it is bundled in with the application runtime, it can be deployed in different environments in which the application runs, maintaining a uniform and ubiquitous security stance.

- **Cisco Duo:** Cisco Secure Access by Duo provides user and device trust (an essential element in protecting cloud workloads). Duo helps to implement a zero trust access model where only authenticated users from trusted devices can access applications, no matter where the application or users are located. It is a key technique to minimize unauthorized access to environments where sensitive data may live and in various cloud environments.

- **Cisco Secure Firewall Cloud Native (SFCN):** Specialized for Kubernetes and containerized serving platforms, SFCN offers next-generation firewall features right inside of Kubernetes. It dynamically scales security services as per the demand, thus enabling nanoscopic network segmentation and policy enforcement and serves as an effective solution to the ephemeral nature of cloud-native applications. SFCN is essential for organizations looking to secure their cloud infrastructure while enabling the flexibility and speed required by DevOps practices.

- **Cisco Secure Workload (**formerly Tetration)**:** As workloads and data move across multicloud and hybrid environments, it is harder to maintain visibility and governance. Cisco Secure Workload can help with this challenge by delivering microsegmentation on the workload. Such a solution allows organizations to apply security policies according to the behavior of workloads and communication between them, thus limiting the attack surface and improving the security posture cloud-native applications achieve over time.
- **Cisco Secure Cloud Analytics (SCA):** When an organization is following a multicloud strategy, it has new requirements for visibility into its cloud security posture. SCA detects potentially intrusive behavior by inspecting network traffic and user activities across cloud stacks. SCA provides deep task detection during cloud usage. It uses forms of machine learning and behavioral modeling to detect outlier actions that show security breaches, insider threats, or compromised workloads and to facilitate the security action frameworks and procedures to respond to suspected security activity swiftly.

When these tools are used in congestion, businesses can achieve consistent security postures, automate security tasks, and maintain compliance with regulatory standards. Cisco's solutions are designed to work cohesively, offering seamless protection that spans the network, applications, users, and data.

Based on what we have discussed so far, securing multicloud and hybrid cloud environments appears to be a daunting but not impossible task, provided that cloud security is approached with the right strategy—consistent policies, expertise with cloud platforms, the right security tools, and the required automation of security practices. By combining these building blocks, any organization can safeguard its cloud resources from the mutable threats landscape. As companies grow more savvy in their adoption of the cloud, the conversation should move from how to use the cloud to how to use it most securely, accompanied by an awareness of the risks and the best practices for addressing them.

Monitoring and Logging Requirements for Compliance

Visibility and transparency are paramount in securing cloud-native environments, where the dynamic and distributed nature of applications—encompassing containers, microservices, and serverless functions—can complicate the monitoring process and obscure key operational insights. These environments' inherent complexity demands robust monitoring and logging practices to maintain effective security oversight.

For this reason, compliance must be integral. It helps to create enforceable security, based on measures that can be verified. It sets minimum standards or rules that define a minimum level of strong security practices the organization will implement.

Compliance sets a basic security baseline, but the controls (which an organization needs to implement to comply) are rarely effective without customizing the control to suit the threat landscape (in terms of risk assessment) of that organization. In other words, compliance regulations dictate a bare minimum that must be met or exceeded, but the scale and scope of controls that get implemented are variable. Prescribed compliance controls may be appropriate across sectors but inadequately reflect a particular organization and the complexities of its operational context; therefore, organizations must review their security controls to both meet compliance and provide effective protection. Such a flexible methodology is essential to not just satisfy compliance standards but to bolster their defenses in a way that is most suitable to their individual security landscape.

Ensuring Visibility and Transparency Across the Cloud-Native Stack

Cloud-native environments have the potential to be complex and distributed to some degree, making it critical to bolster visibility and transparency across the stack of your cloud providers to ensure you maintain a solid security posture. This also opens new doors for advancing security by integrating CSP’s security tools and OpenTelemetry while leveraging third parties. They provide deep insights into the application behavior, system performance, and security threats that can be utilized for monitoring, analysis, and response.

Leveraging OpenTelemetry for Security

The clarity that comes from visibility and transparency is not just a nice-to-have; it is a must. In this maze of challenges, OpenTelemetry—an open-source observability framework that aims to provide robust observability capabilities for cloud-native software—will shine. It provides security teams with the raw APIs and tools they need to collect and export telemetry data like metrics, logs, and traces in a standard way as they are needed. This abundance of data allows for deep insight into behaviors of applications and early detection of security anomalies, such as tracking the flow of data around a service and leveraging time series analysis. OpenTelemetry can quickly detect anomalous behavior that could signal a security breach, such as a DDoS attack indicated by a surge in traffic or abnormal database access patterns indicating a data breach.

When OpenTelemetry is combined with a broad security platform, its value grows even stronger; these integrations enable the correlation between telemetry data and security event data, thereby enabling detection of advanced threats that may otherwise go unnoticed. OpenTelemetry is also an important enabler for security forensics because it can create a detailed contextual account of what happened during every transaction or workflow, where requests originated, and the paths through which they traversed a microservices architecture, and the locations of unauthorized access or attack surface exposure.

These capabilities turn OpenTelemetry into one of the most important weapons in the cloud-native security arsenal, providing more precise approaches to detection of anomalies and assisting in forensic analyses. It increases security visibility but also allows for better threat detection and response strategy, which together reinforce the overall cloud-native security posture to protect against continuously evolving security threats.

Splunk for Enhanced Security Posture

With Cisco's strategic acquisition of Splunk, organizations now have access to a powerful toolset that dramatically enhances their security posture in cloud-native environments. Splunk is arguably one of the most mature data ingestion, correlation, and analytical applications out there; this makes Splunk an absolute powerhouse when it comes to advanced threat detection and security monitoring. It effortlessly ingests and analyzes data, from traditional sources such as network devices, logs, and endpoints, as well as from cloud-native, bleeding-edge telemetry sources. Such large-scale data analysis serves as the basis for more granular threat detection and continuous security monitoring.

Splunk goes even further by allowing users to monitor security-relevant data for live alerts, thereby allowing users to identify threats the moment they occur. It has advanced alerting mechanisms that activate alerts when a certain set of standards is met, allowing the security team to be notified immediately in the event of a potential security incident. Such a system for immediate notification becomes essential for strategies of rapid response that prevent risks in development from snowballing into crises.

Splunk provides deep analysis capabilities and strong integration, and specific capabilities for security analytics. Combining the advanced data analysis and visualization capabilities of Splunk with the comprehensive data collection framework of OpenTelemetry provides organizations with a better understanding of the performance of applications and how secure they are. Such a synergy permits a single pane of glass that is not only descriptive but also prescriptive and actionable in real time.

Additionally, Splunk uses machine learning to provide advanced threat detection and analysis of all the large volume of data collected, such as telemetry from OpenTelemetry, to look for patterns that can indicate a zero-day vulnerability or an advanced persistent threat (APT). With these capabilities, organizations can be proactive in taking action against potential security breaches, which immensely improves their ability to protect the sensitive data and systems needed in a modern, cloud-native environment.

Unquestionably, OpenTelemetry and Splunk are an unbeatable force in the security and observability space for all cloud-native security strategies that a given organization deploys. Together, they strengthen observability of the cloud-native stack and increase the visibility that security operations needs to better protect their environments. The combined capabilities of Splunk's data processing, real-time monitoring, and advanced threat detection, along with OpenTelemetry's observability framework, enables organizations to adopt a more proactive, data-driven approach toward security. Not only does this bring in a level of resilience that is missing in the current threat landscape, but it also bakes a quality defense at the enterprise height.

Continuous Monitoring and Automation Regulatory Requirements and Compliance Standards

Today, in this complex and distributed world of cloud-native computing, regulatory compliance is not about ticking boxes; it's about protecting data and ensuring a basis for trust in the applications that are developed, deployed, and run in this new IT world. Compliance goes well beyond the default security measures, because it requires compliance with a variety of regulatory standards and best practices that are critical for safeguarding data and preserving the integrity of cloud-native systems.

In the disparate world of compliance, automated compliance solutions have been game-changers, shaking off the manual burdens that, pre-AI, came with compliance tasks. These tools do passive, best-effort cleanliness scans on the cloud-native environments and identify and report any deviation from industry standards like PCI DSS, HIPAA, and GDPR. The steady feedback loop delivered by these systems guides actionable responses for immediate correction that optimizes compliance operations with lower risk for breaches or penalties associated with noncompliance.

Policy as Code (PaC), which embeds compliance and security policies into the cloud-native deployment itself, is just as transformative. With policies, organizations ensure that all resources that are deployed will, by default, be compliant with the defined policies, significantly reducing human mistakes that may lead to violations of compliance regulations.

Continuous monitoring and automated compliance solutions can further build the compliance framework. Moreover, bolstered with artificial intelligence and machine learning, these systems are not limited to operational tools' last line of defense in the lightning-speed cloud-native world. By constantly parsing telemetry data to identify immediate threats and enforcing follow-up responses, security teams are more likely to prevent potential incidents from escalating into more serious problems.

An example could be that when a new deployment breaks a compliance requirement, that change can automatically roll back that change in the integrated system, or those systems could also update to comply with the prerequisite standards. This feature resolves the problem quickly and allows you to stay current on the compliance requirements that are constantly changing in a world where regulatory changes are as quick to adapt as the technology itself.

Together, monitoring, logging, and compliance build the framework of cloud-native security that gives you the visibility, control, and assurance of security needed to explore the unknowns of your modern cloud environment. Utilizing modern observability frameworks (such as OpenTelemetry), alongside continuous monitoring and automated compliance solutions, gives enterprises the tools necessary to create robust, secure, and compliant cloud-native applications to ensure response and resilience. This complete perspective tackles near-term security and compliance requirements yet also provides a strong platform for future changes in technology or regulation to adapt, and cloud-native applications can be retained to be secure, compliant, and trusted.

Emerging Trends and Technologies in Cloud-Native Security

The trends and emerging technologies in cloud-native security reflect the larger-scale movement toward more automated, intelligent, and collaborative digital asset protection systems. By utilizing artificial intelligence and machine learning for improved threat detection and response, planning security strategies well in advance for emerging threats with data-driven future-proof capabilities before they can be exploited as zero days, and embracing changing security standards while adopting new frameworks, organizations can ensure that they are better-equipped to maneuver the intricacies of cloud-native security.

The Role of AI and ML in Enhancing Security Postures

AI and ML are transforming cloud-native security with real-time detection and response capabilities. By bringing AI and ML into the fold of cloud-native security, organizations take a giant leap but can then transition away from traditional, reactive, if not outdated security measures and move toward intelligent proactive defenses. Cloud-native security has been significantly affected by AI/ML, not only in automation but in how we detect, analyze, and neutralize security threats.

In cloud-native environments with an exponential increase in the sophistication of data and transactions, common brain-based measures fail because they are not able to keep up with massive incoming data to identify newly emerging threat patterns, and AI and ML are emerging as the key technologies that are being applied and will be applied by businesses to detect threats. Traditional systems use preknown threat signatures, whereas AI and ML explore the ever-changing patterns of crypto threats by scanning the deviations from typical behavior patterns to identify potential threats in real time.

This capability is particularly important in cloud-native settings where the expansive scale can easily overwhelm human analysts. For example, AI-driven systems are trained at detecting unusual activities, such as unexpected data access or transfers that might indicate a data exfiltration attempt, flagging these anomalies well before traditional systems might catch them.

These AI- and ML-driven capabilities go a long way in automating the incident response, which gives a big boost to the agility of the security operations. These systems can respond to threats in real time as soon as they are identified, taking automated and predefined steps to reduce the impact of those threats without human intervention. For example, steps may include quarantining impacted machines, disabling access tokens, or applying automated patches. Also, if abnormal behavior is identified, AI can automatically change firewall configurations to stop any harmful traffic in real time, thus preventing any damage or at least minimizing the attack surface.

In the same vein, one of the most game-changing facets of AI and ML in security is also that they are a piece of predictive analytics. This enhanced method uses past incidents and current developments to anticipate possible future risks, enabling businesses to strengthen their defenses ahead of time. With predictive analytics, the security posture changes from reactive to proactive, and shields your organization from breaches before they take place. If, however, trends indicate an upcoming surge in ransomware attacks on a given family of database systems, AI models can help organizations strengthen their defenses around those core assets ahead of time. This all-hands-on-deck posture is a paradigm shift for the security space, promoting prevention and preparation over reaction, and positions AI and ML as critical weapons in the fight against cloud-native cyber threats.

Enhancing Security with AI/ML: A Practical Cisco Scenario

Consider Cisco’s application of AI/ML in its security operations, demonstrated by TALOS, Cisco’s threat intelligence group. By processing telemetry data across Cisco’s vast infrastructure, TALOS employs AI/ML to spot new threats. With this AI-driven approach, Cisco can grow and deploy protections against emerging threats more quickly. Security is updated continuously to outpace the evolving threat landscape, so defenses are as mobile as the cloud-native apps they protect.

Cisco’s acquisition of Splunk is another example of this security empowerment using AI/ML. While the deal’s size reflects where the market is headed—in this case, cloud-native security—the next evolution marries Splunk’s data-centric architecture with AI/ML for a "predictive security" layer. The ability of Splunk to ingest and analyze massive amounts of data from many sources, including OpenTelemetry, provides organizations with context around the entirety of their security landscapes. AI/ML enhances Splunk’s capabilities, moving it from reactive threat detection towards proactive threat prediction and prevention. It achieves this by integrating complex pattern detection with fine-grained anomaly sensitivity, outperforming traditional solutions.

Together, Cisco and Splunk can shift the paradigm, and this combination accelerates Cisco’s momentum toward increasingly recurring revenue from enhanced digital resilience solutions across the portfolio. Together, Cisco and Splunk are poised to become one of the world’s largest software entities, focusing on securely connecting everything to make anything possible. Together, this integration enhances organizations’ capabilities to manage, protect, and derive the true value of data across ever-growing broken environments of sprawling threat surfaces and multicloud platforms.

In particular, Splunk significantly fits in with Cisco’s current portfolio in that it brings best-of-breed security analytics to Cisco’s broad coverage from devices to applications to clouds. Together, they will unlock observability across hybrid and multicloud environments, enabling customers to provide flawless application experiences essential to driving their digital enterprises. With all the right ingredients for AI, Cisco and Splunk are trusted, scaled and data-embedded to solve these growing challenges and get every organization—big and small—more secure and resilient in the digital world.

Anticipating Future Threats and Preparing Defenses

This integration of AI and ML is not only about detection; it is also about anticipating and managing threats even before they occur. Security teams can leverage the predictive capabilities of AI/ML to identify upcoming threats and weaknesses, which allows them to roll out preventive measures and minimize risks before it is too late. AI/ML + Splunk is the future of cloud-native security in that it delivers automated threat detection, rapid incident response, and preventive analytics. These technologies bring greater speed and accuracy of insights to security teams to allow them to be nimble in responding to a dynamic threat landscape. The potential for AI/ML technologies to enhance the effectiveness of cloud-native security strategies is immense—paving the way for a wealth of innovative defenses—and driving a broader paradigm shift from a reactive security strategy to a dynamic, proactive posture as these technologies mature.

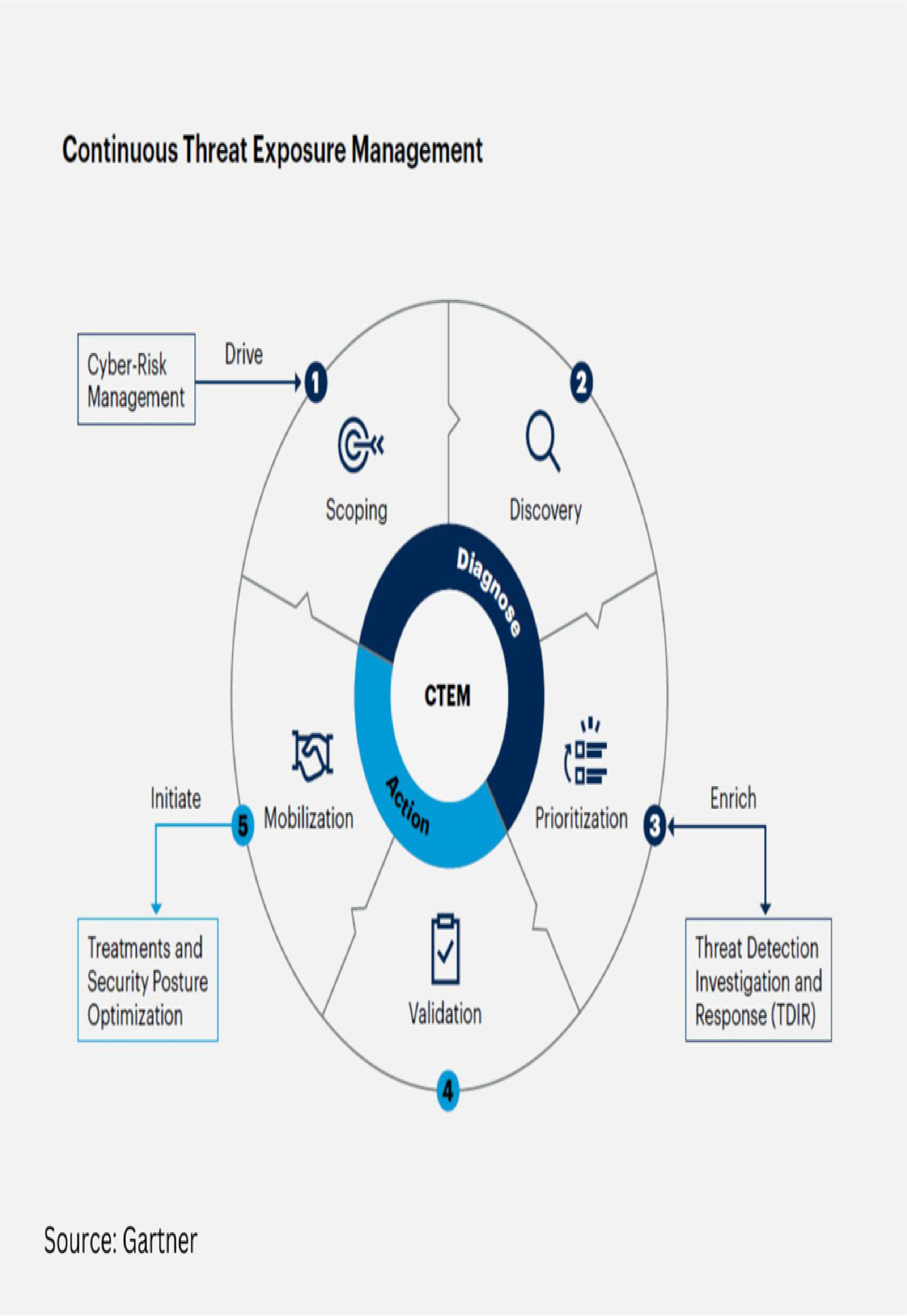
In addition, the overarching strategy for predicting and defending against future threats in cloud-native environments includes drawing on the shared knowledge and teamwork of multiple organizations and security vendors. That shared methodology of threat intelligence sharing is important to detect and develop anticipation on emerging threats. This collective approach allows organizations to broaden their understanding of the cyber threat landscape and to coordinate their defenses more effectively.

Also, the cloud-native paradigm advocates for a continuous reassessment of security postures, using dynamic scanning tools that routinely check for new vulnerabilities and misconfigurations. Ongoing evaluation is essential for staying ahead of rapidly changing cloud innovations and the evolving methods of cyber adversaries. This way, security is not a reactive practice but an ongoing process that adapts to the challenges in a changing environment, and the cloud space remains functional and adaptive against real and future uncouth elements.

Embracing Continuous Threat Exposure Management (CTEM) for Enhanced Cybersecurity

In the field of cybersecurity, conventional vulnerability management (VM) has been described as a largely reactive, haphazard, and labor-intensive exercise—no better than trying to plug holes in a sinking ship. These responses are essential but can be limited in scope to the types of threats and vulnerabilities a typical organization encounters. To meet this new challenge, in 2022 Gartner unveiled the cyber risk management approach of continuous threat exposure management (CTEM) to strengthen cybersecurity defenses in a more holistic, proactive fashion.

Figure 15-6 visualizes the concept of the CTEM framework, which addresses the evolving nature of cybersecurity threats and the necessity for organizations to actively prioritize and manage these threats in a dynamic manner.


Continuous Threat Exposure Management
Cyber-Risk Management
Drive
1
Scoping
2
Discovery
Diagnose
CTEM
Action
3
Prioritization
Enrich
Threat Detection Investigation and Response (TDIR)
4
Validation
5
Mobilization
Initiate
Treatments and Security Posture Optimization

Source: Gartner

Figure 15-6 *Five Steps in the Cycle of Continuous Threat Exposure Management*

The evolution for implementing a CTEM program involves a structured five-step process aimed at enhancing an organization's resilience against cybersecurity threats by continuously assessing the accessibility, exposure, and exploitability of both digital and physical assets.

More Than Vulnerability Management Evolution: Why CTEM Is Gaining Traction

Continuous threat exposure management represents a paradigm shift from traditional VM practices by offering a continuous, systemic approach to identifying, assessing, and mitigating cybersecurity threats and vulnerabilities. Rather than focusing solely on patching known vulnerabilities, CTEM emphasizes a continuous cycle of exposing networks, systems, and assets to simulated attacks. This process helps organizations uncover both known and unknown vulnerabilities, including those that are unpatchable, thereby enabling a more robust and adaptive security posture.

Gartner emphasizes the increasing relevance of CTEM as a vital approach in modern cybersecurity, highlighting several compelling reasons for its growing adoption:

1. **Alignment with Business Objectives:** CTEM works in a tactical way to plan exposure assessment cycles in sync with dedicated business projects or critical threat vectors. This ensures that security measures are not just generic but are specifically designed to protect key business assets and operations, thus aligning security efforts directly with the organization's most critical priorities.
2. **Comprehensive Exposure Management:** Traditional VM programs focus on patchable vulnerabilities; CTEM provides coverage for all exposures (patchable and unpatchable). This holistic method provides a more cohesive insight into an organization skill set fortification, ensuring you highlight all possible weakness as opposed to just those that can be somewhat simply resolved.
3. **Attacker's Perspective and Validation:** CTEM takes an attacker's perspective by focusing on exposures, their validation, and efforts to remediate controls and thereby prioritize exposures with a keen

focus on existing controls that the attacker has bypassed. This technique mimics real-world attacks to verify defenses are in place and that they are configured to stop attacks that have a potential to happen.

4. **Evidence-Based Security Optimizations:** CTEM promotes a shift toward evidence-based security optimizations, which encourages improved collaboration across different teams and shifts the focus from merely tactical responses to strategic, validated enhancements in security. This approach ensures that security measures are not only theoretically effective but are proven in practical deployments.

Getting Started with CTEM

To begin the transition from a legacy VM to a mature CTEM framework, organizations can start by applying CTEM principles to existing risk awareness and management programs. The integration is built around a business-led approach in the sense that it eliminates exposure based on real business value—figuring out what value is behind each specific exposure so that organizations can direct their security efforts to what matters most.

Initial operational wins, which are often achieved through improved prioritization and validation of security findings, can serve as a catalyst for expanding a traditional VM program into a comprehensive CTEM strategy. Embracing cybersecurity validation technologies is crucial in this transition. These technologies enhance prioritization workflows, boost cybersecurity readiness, and ensure that implemented security measures are effective and aligned with business needs. Here is a summary of the five steps required to create a CTEM program:

1. **Scope for Cybersecurity Exposure:** Begin by defining the organization’s "attack surface," which includes all potential entry points and assets vulnerable to attack. This encompasses not only traditional devices and applications but also elements like social media accounts, online code repositories, and integrated supply chain systems.

   A pilot CTEM initiative might target the external attack surface, which has a limited scope but expanding tool ecosystem, or the

SaaS security posture, which is especially relevant because remote work has increased the amount of critical business data being hosted on SaaS platforms.

2. **Develop a Discovery Process for Assets and Their Risk Profiles:** Build out from the scoping to discover all assets, visible and hidden, and their risks, including vulnerabilities, misconfigurations, and more.

   Refrain from the typical trap of a larger number of discovered assets and vulnerabilities equating success and focus on correct scope in accordance with business risk and impact.

3. **Prioritize Threats Most Likely to Be Exploited:** The objective is not to solve every security problem identified, but rather to focus on and prioritize the most urgent threats, and likely to be exploited, considering their urgency, security, whether compensating controls are available, enterprise tolerance of the residual attack surface, and risk to the enterprise.

   Recognize high-value assets as well as develop a strategy plan to tackle the asset threats efficiently.

4. **Validate How Attacks Might Work and How Systems Might React:** Verify whether vulnerabilities are genuinely exploitable, analyze all potential attack pathways, and ensure that the current response plans are sufficiently rapid and robust to protect the organization.

   Secure agreement from all business stakeholders on what specific triggers will lead to remediation actions.

5. **Mobilize People and Processes:** While automated remediation can be useful for addressing certain straightforward security issues, it is crucial to effectively communicate the CTEM plan to both the security team and broader business stakeholders to ensure clarity and understanding.

   The goal of mobilization is to operationalize the CTEM findings by eliminating obstacles to approvals, implementation processes, or mitigation deployments, including the documentation of cross-team approval workflows.

By 2026, organizations that prioritize their security investments based on a continuous exposure management program will be 3x less likely to suffer a breach. This strategic practice not only addresses and minimizes the upfront risk but also fortifies the organization to proactively combat new mandates. For more details, see “Build a Resilient Cybersecurity Roadmap for Your Enterprise” (refer to the “References” section).

Remember, continuous threat exposure management offers a proactive and systemic approach to cybersecurity that aligns security efforts with business objectives while adopting the attacker’s perspective and focusing on evidence-based security optimizations. CTEM empowers organizations to develop a more resilient and adaptive cybersecurity posture as the threat landscape continues to evolve.

The Evolving Landscape of Cloud-Native Security Standards and Framework

The landscape of cloud-native security is continuously evolving, driven by the rapid advancement of technology and the ever-changing nature of threats. Several emerging trends and technologies stand out for their potential to significantly enhance security postures, anticipate future threats, and adapt to new standards and frameworks. A great place for organizations to start navigating this complex terrain is at the core technologies advocated by the Cloud Native Computing Foundation.

The CNCF is critical to the promotion of cloud-native technologies by enabling an open-source ecosystem for securing such applications. By consolidating vital building blocks like Kubernetes, Prometheus, and Envoy and running community events with the latest in security practices and tech, it helps explain this modern way of working so that more companies can take advantage.

A large part of the CNCF’s efforts to promote cloud-native security is to advance and host projects that were created from the ground up for securing cloud-native applications. Sample projects such as SPIFFE and SPIRE tackle identity federation and workload attestation, establishing trust across different systems and organizational boundaries. Similarly, Notary provides crucial mechanisms for signing and verifying container images, enhancing the integrity of containerized applications. Beyond individual projects, CNCF collaborates with industry partners to craft security best practices and compliance standards tailored for cloud-native architectures. These efforts are coordinated through initiatives like the CNCF Security Technical Advisory Group (TAG), which plays a crucial role in defining and disseminating these practices across the industry.

Moreover, the CNCF champions the adoption of open standards in cloud security, which is vital for ensuring interoperability and transparency across various cloud-native technologies. An exemplar of such efforts is the Open Policy Agent (OPA), an open-source engine that enforces unified policies across diverse aspects of the cloud-native stack, from microservices to CI/CD pipelines. This approach not only streamlines security protocols but also supports a flexible yet secure development environment across cloud-native applications.

Put simply, the CNCF, by forging strategic initiatives and projects, literally defines cloud-native security wherein a standardized, open-source, secure echelon is established to cater for the challenges of modern digital operations. As cloud-native technologies continue to evolve, the Security working group will be integral in advocating for strengthened security practices and open standards, while also contributing to the necessary body of tooling that supports well-reasoned approaches to developing secure and resilient digital environments.

Incorporating Matured Zero Trust Frameworks into Cloud-Native Security

By now, it should be evident that as the cloud native security landscape continually evolves, so is the integration of established zero trust frameworks—a crucial step for organizations aiming to strengthen their defense mechanisms. Frameworks developed by the Defense Information Systems Agency (DISA), the Cybersecurity and Infrastructure Security Agency (CISA), and the National Institute of Standards and Technology

(NIST) as an example provide the foundational principles and guidelines to implement zero trust architectures effectively.

DISA Zero Trust Framework

The DISA Zero Trust Framework advocates for a security model where access is denied by default, and every request requires continuous verification, regardless of its origin or destination. This stringent approach is particularly well-suited to cloud native environments, where resources are dynamically scaled and accessed from multiple locations. By integrating DISA's framework, organizations can enforce strict access control and continuous authentication, which are crucial in dynamic cloud environments. Additionally, the framework supports the implementation of segmentation and microsegmentation, which are crucial strategies that isolate resources and minimize the risk of lateral movement within cloud infrastructures.

CISA Zero Trust Maturity Model V2.0

CISA's Zero Trust Maturity Model V2.0 provides a structured roadmap for organizations to evaluate their existing security posture and progressively advance toward a more comprehensive, zero trust–oriented environment. This model's iterative approach complements the ongoing development and deployment cycles characteristic of cloud native systems. It allows for a phased implementation of zero trust principles, which aligns with the cloud native practices of iterative improvement and continuous delivery. The model also enhances resilience against threats through detailed access controls and robust monitoring of cloud resources, ensuring that security measures evolve in tandem with the development environment.

NIST Special Publication 800-207—Zero Trust Architecture

NIST Special Publication 800-207 offers detailed principles and attributes for establishing a zero trust architecture (ZTA), providing a strategic framework tailored for securing IT systems. For cloud-native security, NIST's guidelines are invaluable in designing and deploying applications and infrastructure with security integrated from the outset. The framework promotes a data-centric security model, which ensures that data protection mechanisms are pervasive and persistent across cloud services, reinforcing the security of data irrespective of its location or the environment's complexity.

Cisco Zero Trust Framework

Security is not a one-size-fits-all solution, and zero trust is more than network segmentation. To help understand the architecture, Cisco has broken it down into three pillars, as shown in Figure 15-7:

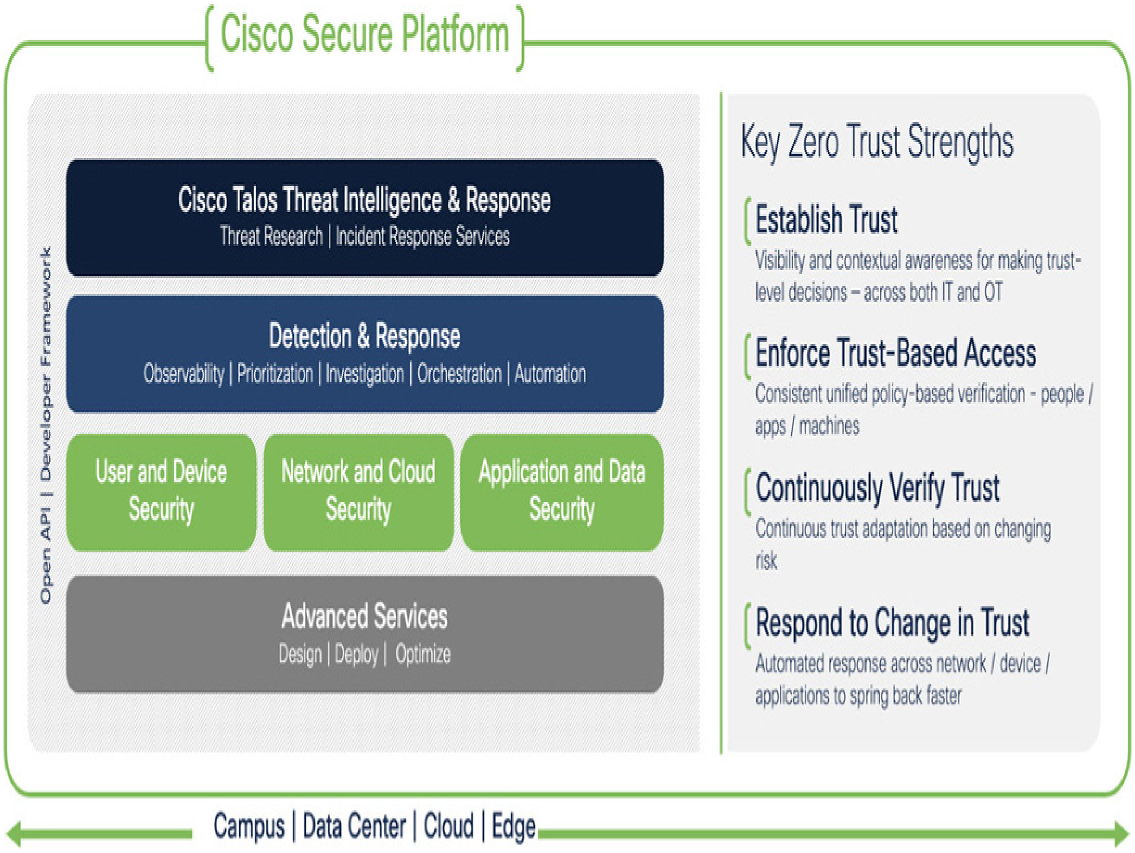

Cisco Secure Platform
Open API | Developer Framework
Cisco Talos Threat Intelligence & Response
Threat Research | Incident Response Services
Detection & Response
Observability | Prioritization | Investigation | Orchestration | Automation
User and Device Security
Network and Cloud Security
Application and Data Security
Advanced Services
Design | Deploy | Optimize
Key Zero Trust Strengths
Establish Trust
Visibility and contextual awareness for making trust-level decisions – across both IT and OT
Enforce Trust-Based Access
Consistent unified policy-based verification - people / apps / machines
Continuously Verify Trust
Continuous trust adaptation based on changing risk
Respond to Change in Trust
Automated response across network / device / applications to spring back faster
Campus | Data Center | Cloud | Edge

Figure 15-7 *Cisco Zero Trust Framework*

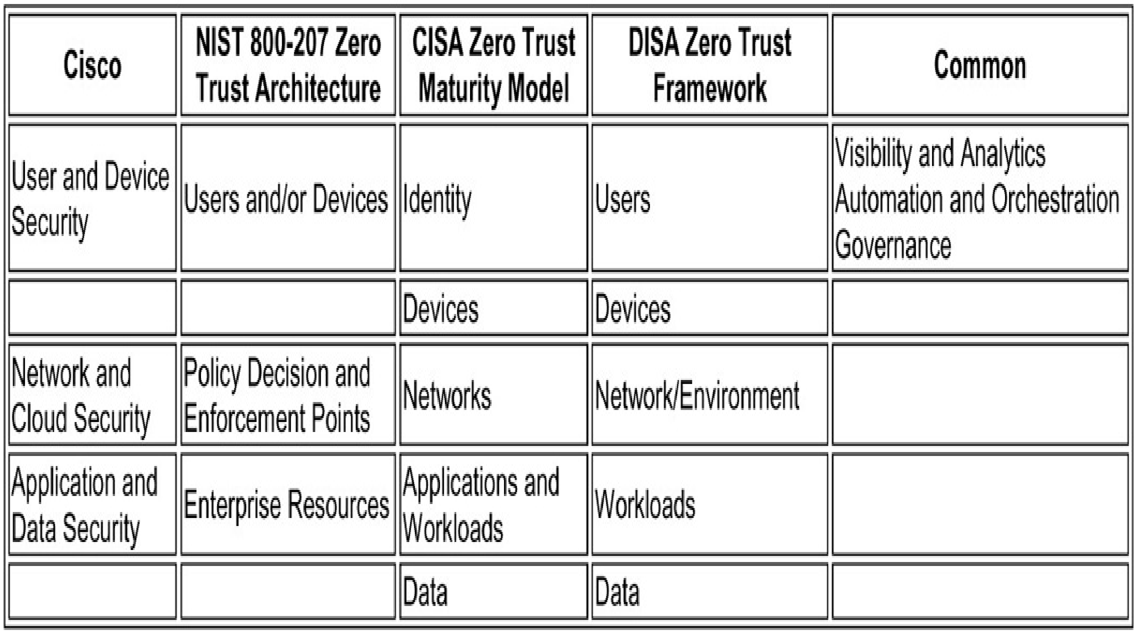
1. **User and Device Security**: Make sure users and devices can be trusted as they access systems, regardless of location.

2. **Network and Cloud Security**: Protect all network resources on-premises and in the cloud, and ensure secure access for all connecting users.

3. **Application and Data Security**: Prevent unauthorized access within application environments irrespective of where they are hosted.

Table 15-3, from Cisco’s “Zero Trust Frameworks Architecture Guide,” shows how zero trust frameworks map to the Cisco Zero Trust Framework (see the “References” section for further details).

Table 15-3 *Zero Trust Security Frameworks Mapping*



Cisco	NIST 800-207 Zero Trust Architecture	CISA Zero Trust Maturity Model	DISA Zero Trust Framework	Common
User and Device Security	Users and/or Devices	Identity	Users	Visibility and Analytics Automation and Orchestration Governance
		Devices	Devices	
Network and Cloud Security	Policy Decision and Enforcement Points	Networks	Network/Environment	
Application and Data Security	Enterprise Resources	Applications and Workloads	Workloads	
		Data	Data	

The Role of Zero Trust Frameworks in Evolving Cloud-Native Security

Zero trust “is also one of the key principles of cloud native security, which encapsulates everything related to dealing with a relatively new sets of technologies allowing advanced feature application while increasing an attack surface” and provides much-needed guidance for organizations looking to improve their safety net in this evermore-open world vacillating on its online infrastructure. These frameworks can allow organizations to incorporate security principles into cloud-native environments, and the technology itself such as microservices, containers, and serverless functions will begin to create a broad security blanket. This ensures that as technology scales and evolves, so do the ways in which security organically grows to meet its changing landscape, while also keeping defense mechanisms serviceable at scale and across variations in underlying systems.

How would this relate to your zero trust strategy? Zero trust principles, in turn, provide a dramatic boost in visibility and control over cloud-native environments—the key enabler for ensuring proper access management and data flow control. This additional layer of monitoring enables organizations to detect and remediate threats more timely, identify inconsistencies in their behavior, and respond to incidents with greater accuracy.

Based on insights from my personal experience (Ariel Leza) of working with various compliance and regulatory communities across the globe, the structure provided by zero trust frameworks can help in meeting the stringent regulatory requirements that we have been trying to harmonize and in creating a well-structured governance model to build a 360-degree compliance strategy. It is also crucial to forming strong governance models, providing a framework for holistic compliances and sustainable strategies.

Zero Trust frameworks provide valuable guidance and principles that enable organizations to not only implement a proactive security posture but also to seamlessly align with the dynamic and elastic nature of cloud-native technologies. These frameworks facilitate the transition from traditional cybersecurity models—often characterized by rigid, ineffective, and isolated security measures—to a more resilient, adaptable, and modern cloud-native architecture. In doing so, organizations are better equipped to face the evolving cybersecurity landscape, transforming their security approach from static, standalone systems into versatile, responsive, and future-ready defenses.

The Cisco SAFE Security Reference Model

The Cisco Secure Architecture for Everyone (SAFE) is an innovative security reference model representing a radically simplified but more robust approach to cybersecurity. SAFE provides a straightforward, building-block approach to tackling the challenges of securing business functions today based on the real-world experience of Cisco's portfolio of security experts, customers, and partners. It simplifies the sometimes complex landscape of cybersecurity into modular pieces, which allows organizations to more easily identify their threats and defenses.

Figure 15-8 provides the key to simplify cybersecurity into secure places in the network (PINs) for infrastructure and secure domains for operational guidance.

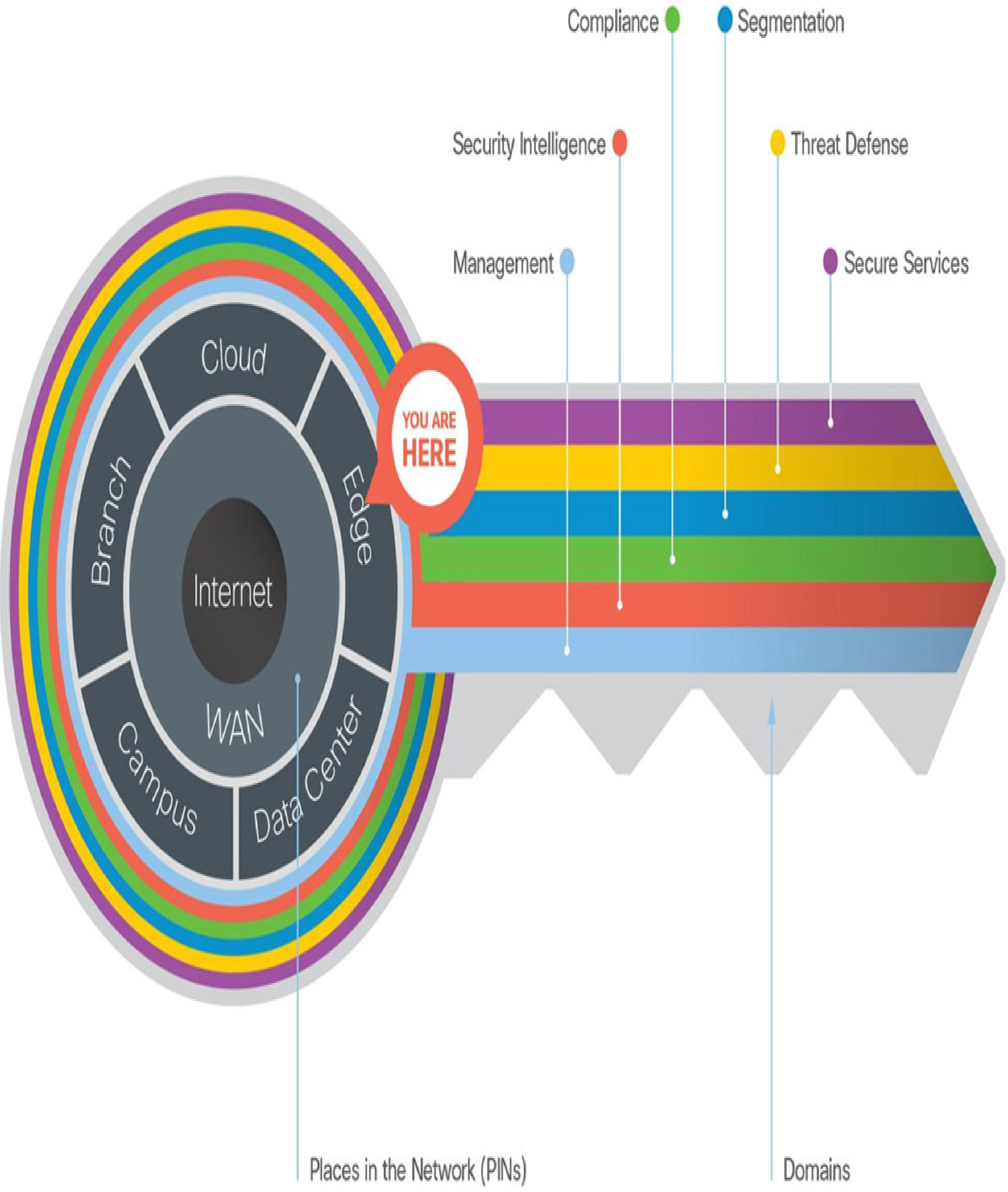


Figure 15-8 *Key to SAFE—The Cisco Security Reference Model*

An overview of this SAFE framework (see the “SAFE Secure Campus Architecture Guide”; refer to the “References” section for more details) is illustrated with a simple but effective visual model: the Key to SAFE, showing secure places in the network (PINs) for infrastructure and secure domains for operational guidance. This systematic classification helps organizations pinpoint the parts of their network and their processes that are crucial for business to protect and provides a structured way to apply security in an orderly manner.

SAFE detects possible vulnerabilities and also provides mappings of security capabilities to various threats faced in different business scenarios. Through this, this model makes threat defense particularly specific and is able to focus on regions most vulnerable to attacks. This information can be used to innovate and create the necessary security strategies ahead of time against any possible breaches that might occur.

These designs provide documented methodologies and best practices for deploying Cisco’s security solutions effectively, ensuring that businesses can implement robust security measures with confidence. Each Cisco Validated Design (CVD) addresses critical security topics, providing a blueprint that organizations can follow to secure their operations against sophisticated threats.

Organizations benefit from in-depth, readily available guidelines and established security product configurations when using Cisco SAFE to protect today’s complex IT environment. When all of this comes together, organizations can improve their security postures in a far more intelligent manner that fits the way the business already works so that those with cybersecurity roles only make things better and never worse.

Summary

Having completed our exhaustive guide to all things cloud-native security, it is clear that securing the elastic, ever-expanding landscapes of the modern world entails a multifaceted strategy. A broad cloud-native security strategy will be more than a defense in depth and involve a large collection of security solutions and perceptions. Often this is more than just CSPM or CWPP but rather extends into IAM or SASE, alongside more traditional segmentation paradigms. They are all essential aspects to enhance the security posture in a layered approach, protecting against the varied spectrum of threats introduced by cloud-native architectures. Furthermore, zero trust permeates all layers of cloud security, which includes user authentication as well as network and workload communications, forming a multiverse of impregnable security. This, alongside existing frameworks—from the likes of DISA, CISA, and NIST, alongside guidelines published by organizations such as the CNCF—ensures that the strategy is both comprehensive and on par with industry standards. As the digital world continues to change, strategies for defending it must change too—and be informed by a desire to closely track the changes and what they mean, a willingness to adopt the latest technologies, and a commitment to working across functions. More than just a strategic decision, it is a pledge to create a safe, agile, and powerful cloud-native environment that understands how to traverse the complex waters of cloud-native security with confidence.

References

1. “Cisco Secure Workload and Secure Firewall White Paper”: https://www.cisco.com/c/en/us/products/collateral/security/secure-workload/sec-workload-firewall-wp.xhtml

2. “Infrastructure as Code Security Cheatsheet”: https://cheatsheetseries.owasp.org/cheatsheets/Infrastructure_as_Code_Security_Cheat_Sheet.xhtml

3. “Foundational Cloud Security with CIS Benchmarks”: https://www.cisecurity.org/insights/blog/foundational-cloud-security-with-cis-benchmarks

4. “Is the Cloud Secure?”: https://www.gartner.com/smarterwithgartner/is-the-cloud-secure

5. “Build a Resilient Cybersecurity Roadmap for Your Enterprise”: https://www.gartner.com/en/cybersecurity/topics/cybersecurity-roadmap

6. “Zero Trust Frameworks Architecture Guide”: https://www.cisco.com/c/en/us/solutions/collateral/enterprise/design

-zone-security/zt-frameworks.xhtml

7. “SAFE Secure Campus Architecture Guide”: https://www.cisco.com/c/en/us/solutions/collateral/enterprise/design-zone-security/safe-secure-campus-architecture-guide.xhtml

Chapter 16. Cloud-Native Application Security

In this chapter, you will learn about the following:

- The unique security challenges of cloud-native applications and how they differ from traditional models.
- The importance of embedding security early in the development lifecycle through DevSecOps and shift-left practices to proactively manage vulnerabilities
- Key CNCF projects that enhance security across Kubernetes, containerized applications, and microservices
- The significance of policy enforcement, vulnerability scanning, and image signing to secure dynamic cloud-native environments
- Best practices for securing serverless architectures, focusing on function-level permissions, API gateway security, and dependency management
- How to protect the software supply chain using a software bill of materials (SBOM), image scanning, and signing tools to ensure the integrity of application components
- The use of Web3 technologies and decentralized identity (DID) for enhanced security and control over user identities in a cloud-native environment
- Large language models (LLMs) leveraged for API security, detecting anomalies and automating policy enforcement

- The implementation of zero trust architectures in cloud-native environments

Introduction to Cloud-Native Application Security

Building on the core principles of cloud-native security covered in Chapter 15, "Cloud-Native Security Foundations" (e.g., scalability, automation, and zero trust), in this chapter we shift focus to how these principles directly affect the security of cloud-native applications—where its modularity, automation needs, and fast deployment cycle create both challenges and opportunities.

Security in cloud-native systems cannot be an afterthought or a layer on top. Rather, it needs to be woven directly into every layer—from the infrastructure running the applications to the application code itself. In this chapter, we will explore the application-specific security driven by the dynamic nature of the interaction between microservices, application programming interfaces (APIs), and containerized environments. We will then focus on the actual mechanisms and methodologies that allow organizations to be able to defend their applications in such decentralized and short-lived spaces.

Definition and Scope

At its core, cloud-native application security revolves around securing applications that are built and deployed using cloud-native principles, such as microservices, containers, and serverless computing. This is a marked departure from traditional security models that often depended on well-defined perimeters. In cloud-native environments, the concept of a fixed perimeter has dissolved, giving rise to a need for fluid and adaptive security practices. These applications are often broken into smaller, independent services (microservices) that operate in highly dynamic, ephemeral environments such as containers or serverless platforms.

In this context, the security strategy needs to evolve—adopting an approach that can handle rapidly changing workloads, dynamic scaling, and continuous integration/continuous deployment (CI/CD). For example, containerized applications may exist for only a few minutes or hours, making traditional monitoring tools ineffective. Securing cloud-native applications, therefore, is about being proactive and integrated, embedding security directly into the development pipeline and runtime environments.

Key Challenges

Cloud-native environments come with their own unique set of security challenges. One of the biggest challenges is the ephemeral nature of the infrastructure. Containers, microservices, and serverless functions are spun up and shut down dynamically, often lasting for only seconds or minutes. This speed demands a shift from traditional, static security approaches to dynamic, real-time monitoring and security enforcement.

Another challenge is ensuring container security. Containers, by their nature, share the host's operating system kernel, which introduces risks of privilege escalation and container breakouts if not properly isolated. Automation, which is essential for scalability in cloud-native environments, adds another layer of complexity. Security teams must now rely on automated tools and pipelines to enforce security policies without human intervention, ensuring that applications remain secure at scale.

Lastly, scaling security practices to accommodate the massive, distributed nature of cloud-native applications can be daunting. The complexity grows with the number of microservices, the volume of API interactions, and the increased attack surface that comes with CI/CD pipelines.

Evolution from Traditional Security

This transition from traditional monolith architectures to cloud-native designs is a massive shift in how applications are developed and secured. In previous paradigms, the approaches to security were designed for environments that were much more static and thus predictable. Firewalls, intrusion detection systems (IDS), and endpoint security were effective defenses in the old world where applications operated within isolated, stable environments and were hosted on predictable, co-located infrastructure.

Cloud-native architectures introduce a more dynamic paradigm—made up of microservices, usually running in ephemeral containers that communicate via APIs, which mandates a security strategy that can continuously align with changing workloads and distributed components. As this new environment matures, protection needs to be embedded outside of traditional perimeters and integrated into the lifecycle of applications from development to runtime.

In cloud-native environments, microservices are designed to be small, independently deployable units that communicate via APIs. Containers package these services into isolated units but also introduce new attack surfaces, such as vulnerabilities in container images, runtime environments, and orchestration platforms like Kubernetes (K8s). These complexities necessitate security models that are integrated into every layer, from infrastructure to application code, using principles like zero trust, least-privilege access, and continuous monitoring.

Understanding OWASP and Cloud-Native Security Risks

The Open Worldwide Application Security Project (OWASP) has long served as a foundation for application development and security, outlining the most critical vulnerabilities and their solutions. This regularly updated resource has been helping organizations tackle the risk of web applications for diverse architectures and technologies since the first version of the OWASP Top 10 risk list was released in 2003. In April 2022, OWASP put together the Cloud-Native Application Security Top 10 to represent the unique challenges to securing cloud-native environments.

Although both lists are rooted in fundamental security and threat concepts, the latter also includes risks and threats that arise from the distribution complexity of cloud-native systems, automated CI/CD pipelines and short-lived workloads. To illustrate, risks such as CI/CD Pipeline and Software Supply Chain Flaws and Inadequate ‘Compute’ Resource Quota Limits are limited to cloud-native contexts, indicating the need for familiarization with the architecture-related concerns in these environments.

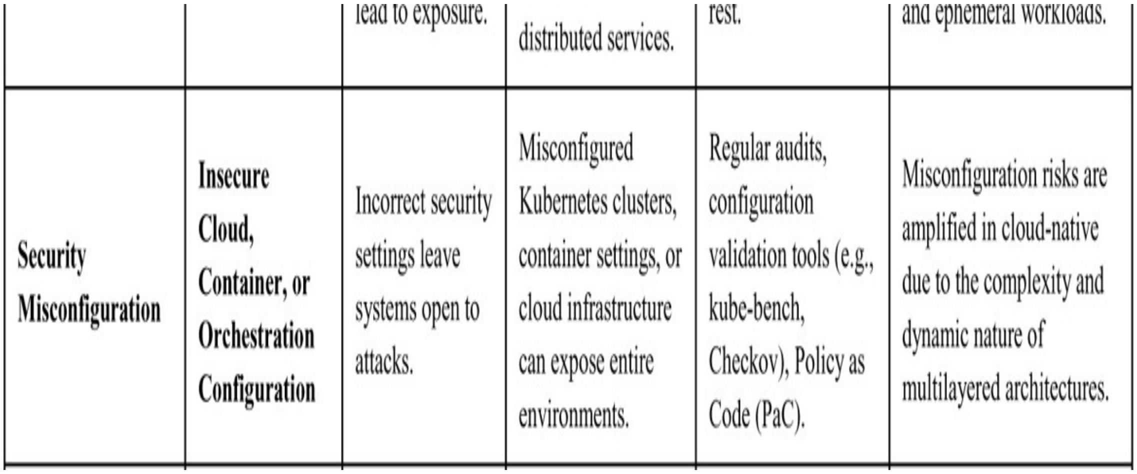
OWASP Top 10: Web vs. Cloud-Native

Table 16-1 compares different types of risks from both the OWASP Top 10 Web Application Risks and the Top 10 Cloud-Native Application Security Risks, emphasizing their relevance and differences in the cloud-native context. This table is divided into the following columns:

- **OWASP Top 10 Web App Security Risk:** Discusses the OWASP Top 10 for web app vulnerabilities, the traditional focus of the OWASP Top 10.
- **OWASP Top 10 CN App Security Risk:** Lists risk from Cloud-Native Application Security Top 10, targeting to Cloud-native architectures.
- **Description:** Provides a high-level description of what the risk is and the impact it has globally.
- **Cloud-Native Context:** Describes how this risk is different in a cloud-native environment and what specific challenges it brings.
- **Mitigations and Tools:** Provides some tools or frameworks to mitigate the risk.
- **Similarity or Uniqueness in Cloud-Native Context:** Compares the overlap or uniqueness of the risk between the two lists. It distinguishes whether this threat is simply the same as existing risk in web applications, or whether this risk is uniquely amplified in cloud-native architectures.

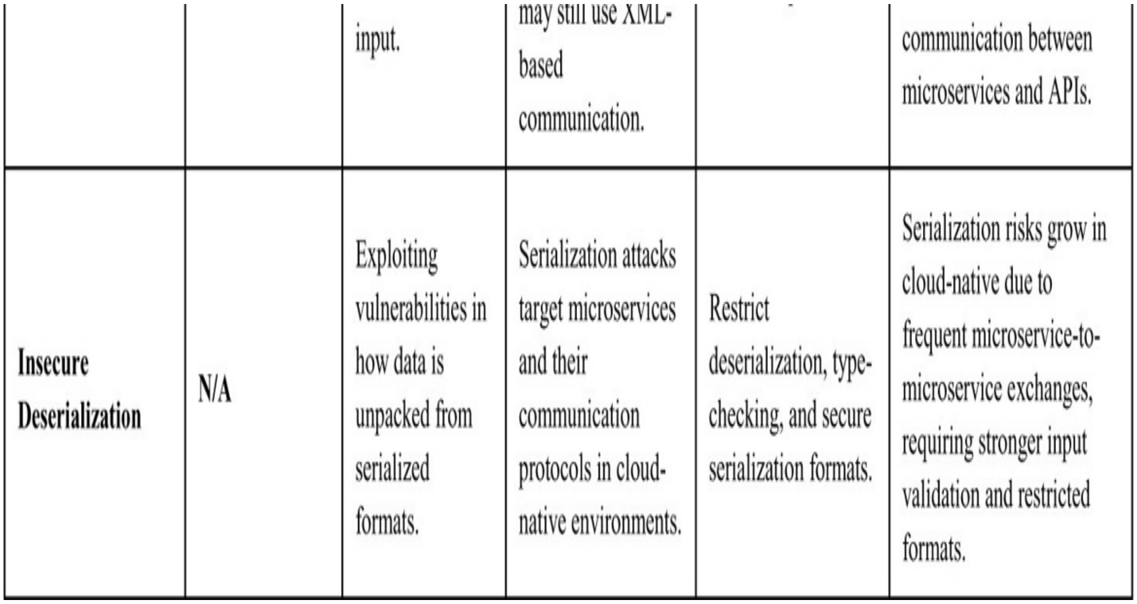
Table 16-1 *OWASP Top 10 Security Risks for Cloud-Native Environments*

OWASP Top 10 Web App Security Risk	OWASP Top 10 CN Application Security Risk	Description	Cloud-Native Context	Mitigations and Tools	Similarity or Uniqueness in Cloud-Native Context
Injection Attacks	**Injection Flaws (App Layer, Cloud Events, Cloud Services)**	Untrusted data sent to interpreters, such as SQL or NoSQL, potentially compromising systems.	API-heavy communication between microservices increases the risk of injection attacks in cloud-native systems, including Kubernetes APIs and cloud event triggers.	Input validation, API gateways, policy enforcement tools (e.g., OPA, Falco).	Similar risk, but cloud-native introduces injection avenues via cloud services and event-driven architectures. These platforms rely heavily on APIs, expanding the attack surface.
Broken Authentication	**Improper Authentication and Authorization**	Weak or misconfigured authentication systems allow unauthorized access.	Managing authentication across distributed systems, including APIs, containers, and serverless functions, adds complexity in cloud-native.	Enforce IAM policies; use tools like Kyverno, MFA, and API Gateway authentication.	Cloud-native's distributed nature amplifies challenges, requiring service-to-service authentication and authorization controls. Dynamic scaling also adds complexity.
Sensitive Data Exposure	**Insecure Secrets Storage**	Failure to secure sensitive data, including passwords and API keys, can	Cloud-native environments often deal with secrets shared across ephemeral workloads and	Secure secrets management (e.g., Vault, AWS Secrets Manager), encrypt data in transit and at	Traditional and cloud-native overlap in securing sensitive data, but the latter demands scalable secrets management for dynamic



		lead to exposure.	distributed services.	rest.	and ephemeral workloads.
Security Misconfiguration	**Insecure Cloud, Container, or Orchestration Configuration**	Incorrect security settings leave systems open to attacks.	Misconfigured Kubernetes clusters, container settings, or cloud infrastructure can expose entire environments.	Regular audits, configuration validation tools (e.g., kube-bench, Checkov), Policy as Code (PaC).	Misconfiguration risks are amplified in cloud-native due to the complexity and dynamic nature of multilayered architectures.

Using Components with Known Vulnerabilities	**Using Components with Known Vulnerabilities**	Third-party libraries and container images with security flaws can be exploited.	Cloud-native apps rely heavily on container images, third-party libraries, and dependencies, increasing the potential attack surface.	Regular updates, dependency scanning (e.g., Trivy), image signing tools like Notary.	While this risk exists in both contexts, cloud-native environments are more vulnerable due to their dependency-heavy nature and reliance on containerized infrastructure.
Insufficient Logging and Monitoring	**Ineffective Logging and Monitoring (e.g., Runtime Activity)**	Lack of visibility into security incidents allows breaches to go undetected.	Cloud-native environments introduce challenges with distributed and ephemeral resources, making logging and monitoring more complex.	Logging tools (e.g., Fluentd, Prometheus), real-time monitoring (e.g., Falco, AWS CloudWatch).	Monitoring in cloud-native must aggregate logs across distributed components like Kubernetes clusters, API gateways, and serverless functions, creating unique challenges.
Cross-Site Scripting (XSS)	**N/A**	Attackers inject malicious scripts into web pages viewed by users.	Cloud-native apps with web-based UIs remain vulnerable to XSS if input isn't properly sanitized.	Use frameworks like ReactJS, input validation, and escaping untrusted HTTP requests.	This is more common in traditional web applications but persists in cloud-native when web interfaces interact with APIs and microservices.
XML External Entities (XEE)	**N/A**	Exploiting vulnerable XML parsers through malicious XML	Cloud-native apps often avoid XML in favor of JSON, but legacy systems or hybrid environments	Switch to JSON, secure XML parsers, sanitize input data.	XEE risks are less prominent in cloud-native architectures due to the widespread adoption of JSON over XML for



		input.	may still use XML-based communication.		communication between microservices and APIs.
Insecure Deserialization	**N/A**	Exploiting vulnerabilities in how data is unpacked from serialized formats.	Serialization attacks target microservices and their communication protocols in cloud-native environments.	Restrict deserialization, type-checking, and secure serialization formats.	Serialization risks grow in cloud-native due to frequent microservice-to-microservice exchanges, requiring stronger input validation and restricted formats.

N/A	**CI/CD Pipeline and Software Supply Chain Flaws**	Vulnerabilities in CI/CD pipelines or tampered dependencies compromise security.	Cloud-native environments rely heavily on automated CI/CD pipelines, making supply chain security critical.	Signed artifacts (e.g., Notary), dependency scanning tools like Checkov, secure software updates with TUF.	Unique to cloud-native due to the prevalence of automated pipelines and significant reliance on third-party tools and containerized dependencies.
N/A	**Over-Permissive or Insecure Network Policies**	Improper segmentation or overly permissive policies enable lateral movement.	Dynamic network configurations in Kubernetes and multicloud setups require strict segmentation and policy enforcement.	Network policies (e.g., Calico), service meshes, microsegmentation.	Unique to cloud-native architectures due to their reliance on interconnected services and dynamic, ephemeral networks.
N/A	**Improper Asset Management**	Difficulty in tracking and managing ephemeral cloud-native resources.	Cloud-native environments use dynamic resource creation, making traditional asset tracking tools inadequate.	Automated discovery tools, tagging, Kubernetes metadata analysis.	Cloud-native introduces ephemeral resources and multicloud setups, requiring automated and dynamic asset management solutions.
N/A	**Inadequate "Compute" Resource Quota Limits**	Failing to set resource quotas can lead to resource exhaustion or denial-of-service (DoS) attacks.	Kubernetes and cloud-native workloads require resource quotas to prevent excessive resource usage or abuse.	Quota enforcement, resource limits in Kubernetes, monitoring resource utilization.	Unique to cloud-native environments, where orchestration platforms like Kubernetes dynamically allocate resources, making quotas and monitoring critical for stability and security

					security.

Expanding OWASP Principles for Cloud-Native Architectures

Although OWASP original guidelines are tailored for classic web applications, they still form a cornerstone to identify security risks while still applicable to modern cloud-native environments. Cloud-native architectures bring in new challenges: ephemeral infrastructure, microservices, containerized workloads, and orchestration tools like Kubernetes, which help automate much of the management process but also add to the complexity. Addressing these complexities by extending OWASP principles is essential to securing distributed and scalable systems in a proper and best-practice way.

Injection and Misconfiguration Risks in Cloud-Native Contexts

Injection attacks such as SQL injection, NoSQL injection, and command injection thrive in cloud-native environments where the reliance on APIs, service meshes, and cloud event triggers only exacerbates the insecurity. Infrastructure layers create additional exposure within environments (e.g., Kubernetes, Helm chart misconfigurations).

To mitigate any risk of injection, organizations should take advantage of schema validation frameworks for the APIs, which have strict content sanitization policies to guarantee that, if malicious data crosses the system boundary, the malicious content will not propagate though the system. Misconfigurations, such as open ports or over-permissive IAM roles, can be mitigated by automating security baselines with tools that validate Infrastructure as Code (IaC) templates and runtime configurations. For example, Policy as Code (PaC) frameworks can eliminate manual errors and the resultant exposure by enforcing secure configurations at the time of deployment.

Authentication and Access Control in Distributed Systems

Service-to-service communication and user interactions in cloud-native environments need highly granular authentication and access control mechanisms in place to be secure. These architectures often employ ephemeral resources and distributed microservices that run for a short duration, and therefore, they require federated identity management and dynamic access controls, which are different from traditional applications.

Federated identity management systems maintain multicloud and hybrid deployment seamlessness by ensuring that users and services are authenticated in a consistent manner. To enhance security, dynamic role binding provides temporary access privileges and scopes roles to certain attributes or sessions, thus shrinking the window of opportunity for unauthorized access.

To further mitigate risks, automated secrets rotation is essential in cloud-native environments to prevent the use of static credentials from being leaked and exploited. When combined with workload-specific policies, these methods can greatly enhance the security posture of microservices and containers.

Managing Component Vulnerabilities at Scale

Cloud-native systems are more modular and make extensive use of third-party libraries, containerized dependencies, and external APIs. Each fragmented section brings risks of its own, and averting exploitable weaknesses within a connected universe of interacting microservices demands a strong observability operation.

A dependency graph security approach captures the relationship between services, libraries, and container images. This allows teams to monitor vulnerabilities in real time. When you have such an enterprise environment, it can be a large architecture that can be distributed as well. Also, teams can implement software bill of materials (SBOM) systems that make it possible to track dependencies while preventing the use of known vulnerable components.

With an automated patching pipeline and continuous vulnerability feed, any deprecated or vulnerable components are patched seamlessly with a secure software development lifecycle.

Cloud-Native Logging and Monitoring for Security

The distributed nature of cloud-native systems generates vast volumes of logs and metrics, complicating traditional monitoring techniques. Context-aware observability tools are essential in this landscape, because they correlate logs and metrics with specific resources or workloads, offering actionable insights. For instance, ephemeral trace analysis can capture and analyze security incidents in transient resources like containers or serverless functions, ensuring visibility even in fleeting workloads.

Policy-driven alerts dynamically adjust thresholds based on workload profiles, reducing noise and highlighting actionable threats. These approaches address the inherent challenges of monitoring cloud-native workloads, ensuring visibility, and enabling quick responses in highly dynamic environments.

Should We Consider a Cloud-Native-Specific Model?

While OWASP offers a proven foundation for application security, the complexities of the cloud-native landscape present new challenges beyond traditional web security. The challenges addressed by frameworks such as those from the Cloud Native Computing Foundation (CNCF) are met with tools specifically suited to cloud-native architectures.

A hybrid approach is when we implement these CNCF tools along with OWASP principles, which helps us ensure that we are adopting both the time-proven practices along with the recent advancements specific to the cloud-native paradigm. This strategy tackles both the application layer and the underlying infrastructure to ensure robust security.

CNCF Projects for Cloud-Native Security

CNCF has an entire ecosystem of tools to take care of cloud-native security challenges. There are six core categories of these tools, and within those categories, subcategories that showcase functionality and capabilities. We’ll explore each of these categories and the security capabilities they facilitate and end with a table of some of the key tools used in each of the categories and how they are used.

For a more detailed guide of the categories and their scope of problem-resolution and project’s focus, refer to the “CNCF Official Guide” (see the “References” section at the end of this chapter.)

1. Provisioning

The provisioning category includes tools that establish the foundation for a secure cloud-native environment. They include subcategories like infra automation, security, and compliance enforcement. These services help organizations standardize the creation of resources, enforce policies, and enforce zero trust practices at the time of provisioning.

Infrastructure Automation

By injecting security guardrails directly into provisioning workflows, organizations can automatically create resources using IaC. Doing so minimizes human error, increases consistency, and enables scalability, which allows secure environments to be deployed instantly across different cloud platforms.

Security and Compliance Enforcement

Compliance auditing and policy enforcement mechanisms also are included in provisioning tools. The tools are in the area of auditing Kubernetes clusters and other infrastructure for standards such as CIS benchmarks, as well as documentation that supports compliance with regulatory requirements. Furthermore, they align resource creation with zero trust by automating compliance and preventing misconfiguration using security policy engines and identity management solutions.

2. Runtime

The Runtime category focuses on safeguarding workloads and applications during production. Cloud-native applications rely on a secure operating foundation using runtime tools, which builds your business stronger, eliminating threats dynamically as they emerge.

Runtime security monitoring and confidential computing are subcategories, both answering the challenges associated with the dynamic nature of cloud-native environments by adding extra layers of protection.

Runtime Security Monitoring

Runtime tools detect anomalous behaviors, such as privilege escalation or unauthorized access, in real time. These solutions complement provisioning-stage security by continuously monitoring workloads for threats during execution. By identifying deviations from expected behavior, runtime monitoring enhances the resilience of cloud-native architectures.

Runtime tools identify abnormal behaviors in real time, such as privilege escalation or unauthorized access. While they serve as a great supplement to provisioning-stage security, they are focused on threat detection during the execution of your workload. In cloud-native architectures, runtime monitoring helps build resilience using deviations from expected behavior.

Confidential Computing

Confidential computing protects sensitive workloads—even when running—by isolating them from malicious actors. These tools ensure that workloads are safely isolated from one another, even in untrusted execution environments, with robust isolation mechanisms available in lightweight container runtimes. This also reinforces the zero trust principle down to the hardware level with hardware isolation empowered by a trusted execution environment (TEE).

3. Orchestration and Management

Tools in the Orchestration and Management category manage distributed cloud-native environments with plenty of security controls in place, primarily Kubernetes, to form the backbone of cloud-native environments. The extra security layers offered by these tools allow organizations to enable the secure build, scale, and operation of complex, distributed applications, all while avoiding operational headaches. These tools can be policy enforcement, service meshes, and API gateways and proxies—classes of tools that are used to secure and simplify the operation of Kubernetes clusters and microservices architectures.

Policy Enforcement

Policy enforcement tools help Kubernetes clusters and microservices architectures comply with security policies and work within certain compliance frameworks. They automate compliance across environments and ensure infrastructure operations are aligned with organizational and regulatory standards.

Service Meshes

Service meshes make it easy to communicate between microservices securely and transparently. They have capabilities like mutual TLS (mTLS), fine-grained access control, and traffic management for secure communication between services without adding extra application-layer complexity.

API Gateways and Proxies

API gateways and proxies help impose security and traffic control on internal and external APIs. They secure zero trust access by enforcing authentication and authorization rules to prevent unauthorized access to APIs and protect against any communication or data exposure.

4. App Definition and Development

The App Definition and Development category secures the entire lifecycle of application development, deployment, and the software supply chain. Particular subcategories like Image Security and Supply Chain Security are fundamental for protecting the authenticity and integrity of container images, protecting software updates from tampering, and helping organizations to detect vulnerabilities early in the development process. This approach of ensuring that each element is validated before deployment and the risks are managed throughout its lifecycle are aligned with zero trust principles.

Image Security

You need to ensure that any container images are validated before they enter production and, therefore, authenticated or that tampered artifacts are not running in production. There are tools that do image signing, like Notary, to make sure that a specific container image has not been tampered with and comes from a trusted source. Image validation provides additional assurance that only compatible containers will be deployed, which helps reduce the possibility of deploying compromised components and contributes to a strong supply chain security posture.

Supply Chain Security

Securing software updates from any type of meddling is vital to software supply chain security. To mitigate against these risks, The Update Framework (TUF) guarantees that updates are both signed and, importantly, verified to protect against unsanctioned changes or tampering. Furthermore, early detection tools for vulnerabilities integrated into CI/CD pipelines help find risks during development so a proactive approach to supply chain security can be taken.

Integrating Security into CI/CD Pipelines

The incorporation of security in CI/CD pipelines enables organizations to discover vulnerabilities early in the software delivery lifecycle and promote the use of validated and signed artifacts. By incorporating security scanners and vulnerability detection frameworks during the development phase, developers save time, and risks are minimized before deployment. By integrating security earlier in the continuous development pipeline, it strengthens security throughout the application lifecycle and aligns with zero trust by ensuring only approved and validated components are deployed.

5. Observability and Analysis

The Observability and Analysis category lifts the gaze across cloud-native environments for proactive security monitoring, incident detection, and response.

These tools give visibility into how apps run and make use of resources, allowing for the rapid detection of anomalies, breaches, and performance bottlenecks. This can be further subcategorized into log and metrics monitoring—where resource consumption and application health is tracked

—and distributed tracing, which provides end-to-end visibility of request flows through multiple services.

Observability is indeed a key pillar of zero trust driven by the need for proper data so that an organization can continuously monitor, validate, and improve workloads and infrastructure security.

6. Platforms

The Platforms category provides integrated solutions that package together tools and functionalities to ease the adoption and running of cloud-native technologies Platforms abstract away the complexity and overhead of working with individual tools so organizations can deliver and scale apps securely without having to manage the infrastructure behind them.

Weaving security into the architecture is important to cloud-native security; platform makes a difference. Examples of such practices are role-based access controls (RBACs), encryption, policy enforcement, compliance audit, and secure multitenancy, guaranteeing the isolation of workloads, which is an essential need of zero trust architectures. Platforms provide a seamless combination of different functionalities, reducing operational overhead and providing a one-stop-shop for security that encompasses the deployment, orchestration, and monitoring layers.

In traditional software, the value is obvious with platforms, but cloud-native platforms take it one step further by enabling consistent multicloud strategy across the organization. These solutions have innate features to leverage zero trust, where no user, device, or service is trusted by default and each action must be verified.

From a security perspective, these platforms provide centralized identity management, constant policy enforcement, workload isolation, and encryption, serving as the foundation for these very platforms. They also help to simplify regulatory compliance, automating checks and providing visibility into configurations, which are key for organizations trying to navigate complex security and compliance standards.

Security Platforms in the Cloud-Native Ecosystem

There are countless projects, listed on the CNCF landscape, that tackle particular pieces of cloud-native security, such as policy enforcement, runtime protection, and workload identity. That said, fully integrated open-source security platforms that bring these capabilities together into a single solution are lacking.

As K8s security posture management (KSPM) tools and vendor-backed platforms built to provide holistic security across cloud-native environments have emerged, they have all spotted this gap and decided to address it.

Examples of KSPM Tools

Tools that fall under the category of KSPM are about visibility and management of Kubernetes environments for security and compliance purposes. This problem is compounded by the fact that these tools target specific Kubernetes-related risks, but are most often siloed within the various disciplines, forcing teams to put in sweat equity just to pull together a comprehensive security strategy. Examples include

- **Kubescape (Open Source):** A security tool for Kubernetes to help you scan your clusters for misconfigurations and run compliance based on industry benchmarks such as CIS Kubernetes
- **Kube-bench:** An open-source tool for reviewing Kubernetes clusters against the CIS Kubernetes Benchmark to find misconfiguration
- **Checkov (Open Source):** An IaC scanning tool that analyzes Kubernetes manifests to find potential security risks

The Role of Vendor Platforms

The absence of truly integrated open-source security platforms has led many vendors to take a combined approach, unifying existing cloud-native tooling with strong integrations, simplicity of operations, and more broadly, advanced security capabilities. These vendor platforms enhance the functionalities of CNCF projects by integrating them within centralized operational environments and providing the connective tissue between disparate technologies, infrastructure, and applications.

For example, Cisco’s cloud-native application security solutions, among other CNAPP solutions integrate CNCF projects and best-in-class tools to deliver three key security functionalities:

- **Container and Kubernetes Security Posture Management**: This functionality encompasses runtime protection, policy enforcement, and compliance auditing.
- **Zero Trust Capabilities**: This includes fine-grained identity verification and workload isolation, essential for securing cloud-native applications.
- **Supply Chain Security**: This provides critical features such as software bill of materials (SBOM) and container scanning to defend against supply chain threats.

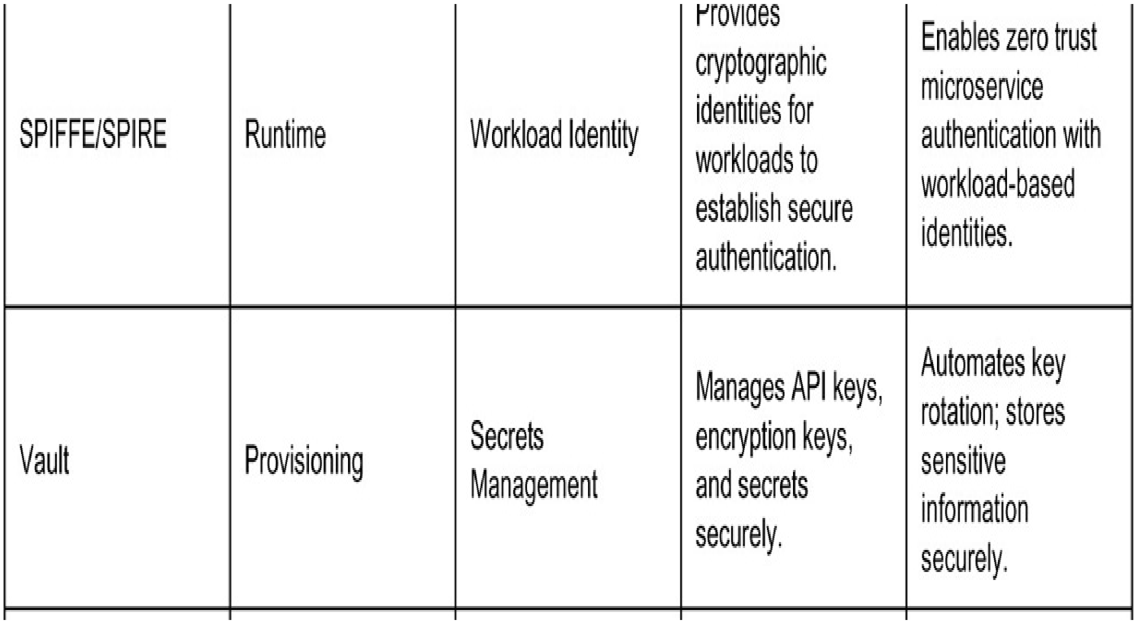
Vendor platforms hide complexity, automate workflows, and ensure uniform security across hybrid and multicloud environments. They allow organizations to implement unified security strategies with the scalability and operational simplicity of using cloud-native tools.

Top 20 CNCF Projects Focused on Security

The projects included in Table 16-2 were selected for their direct relevance to addressing critical aspects of cloud-native security (for example, runtime protection, policy enforcement, compliance, identity and access management, and API security). The principles and features of these tools have some strong alignment with zero trust because they focus on securing dynamic and distributed environments, protecting against modern threats, and ensuring workloads and data integrity, confidentiality, and availability. Each project contributes to building a comprehensive security strategy for cloud-native architectures. Often organizations will choose to adopt several tools and integrate them for a consistent and more modernized operational model and framework support.

Table 16-2 *CNCF Projects for Cloud-Native Security*

CNCF Project	Category	Type of Solution	Functionality	Use Cases
Kyverno	Orchestration and Management	Policy Enforcement	Provides Kubernetes-native policy engine for enforcing security policies.	Enforces non-root execution, validates secure registries, automates cluster-level policy checks.
Falco	Runtime	Runtime Security	Detects anomalous behavior by monitoring system calls.	Provides real-time threat detection, privilege escalation prevention, unauthorized file access alerts.
Notary	App Definition and Development	Image Integrity and Authenticity	Ensures only signed, trusted container images are deployed.	Prevents the use of tampered or unverified container images.
TUF	App Definition and Development	Software Supply Chain Security	Protects against tampering during software updates.	Secures update mechanisms to mitigate supply chain vulnerabilities.
OPA	Provisioning and Management	Policy Engine	Provides context-aware policy enforcement for workloads and infrastructure.	Implements fine-grained access controls; enforces zero trust policies across cloud resources.



SPIFFE/SPIRE	Runtime	Workload Identity	Provides cryptographic identities for workloads to establish secure authentication.	Enables zero trust microservice authentication with workload-based identities.
Vault	Provisioning	Secrets Management	Manages API keys, encryption keys, and secrets securely.	Automates key rotation; stores sensitive information securely.

Keycloak	Platforms	Identity and Access Management (SSO)	Simplifies authentication and authorization for applications.	Provides single sign-on (SSO), federated identity management, and user authentication.
KubeBench	Provisioning	Compliance Tool	Audits Kubernetes clusters to ensure adherence to CIS benchmarks.	Identifies misconfigurations; enforces compliance with security best practices.
Trivy	App Definition and Development	Vulnerability Scanning	Scans container images and file systems for vulnerabilities.	Identifies common vulnerabilities and exposures (CVEs) and misconfigurations during CI/CD workflows.
Calico	Orchestration and Management	Networking and Network Security	Implements network policies and security controls for Kubernetes.	Enforces zero trust network segmentation; provides microservice communication visibility.
Cilium	Orchestration and Management	Network Security	Offers eBPF-based networking and security for Kubernetes	Enables fine-grained network policies, traffic encryption, and transparent

			workloads.	transparent observability.
Sysdig	Observability and Analysis	Runtime Security Monitoring	Provides deep visibility into container and Kubernetes runtime environments.	Detects security incidents, enforces compliance, and performs forensics on runtime threats.
Kubescape	Provisioning	Kubernetes Security Compliance	Ensures Kubernetes manifests meet security and compliance standards.	Validates configurations against best practices and compliance frameworks like NSA and CISA.
BPFMan	Runtime	Runtime Security and Forensics	Provides eBPF-based monitoring and runtime analysis for identifying threats and anomalies.	Offers in-depth forensics, detects runtime attacks, and provides visibility into workload activity.

APIClarity	Observability and Analysis	API Security and Monitoring	Provides visibility into API traffic to detect shadow APIs and anomalies.	Detects misbehaving APIs, identifies unapproved API usage, and enforces API security policies.
Airlock	Runtime	Runtime Application Defense	Enforces runtime protection by isolating sensitive application components.	Protects applications against injection attacks, unauthorized access, and other runtime threats.
Aserto	Platforms	Authorization and Access Control	Enables fine-grained, real-time authorization for applications and APIs.	Implements zero trust access control; enforces policies across applications and user requests.
Confidential Containers	Runtime	Confidential Computing	Secures container workloads using trusted execution environments (TEEs).	Protects sensitive data during processing, ensuring workloads run securely in isolated environments.
Checkov	App Definition and Development	Infrastructure-as-Code Security	Scans IaC files, including Kubernetes manifests, for security risks	Identifies vulnerabilities and misconfigurations early in the development

			security risks.	lifecycle.

For more information, refer to “CNCF Landscape” (see the “References” section at the end of this chapter).

The 4C’s of Cloud-Native Security—Cloud, Clusters, Containers, and Code—provide a structured framework for securing applications across the layers of a cloud-native stack. Each *C* addresses a distinct aspect of the environment, demanding tailored security controls to ensure comprehensive protection. Many of the concepts mentioned beforehand are elaborated throughout this book in detail, but it is important to refer to these concepts as part of the explanation of the 4C’s.

Cloud

The Cloud layer represents the lowest level at which apps reside, providing the first line of defense. Among the most daunting challenges organizations face is how to standardize identity and access management (IAM) across what can be the disparate configurations and capabilities of hybrid or multicloud environments. Such complexity can lead to either unsynchronized permissions or excessive privileges on accounts. As a solution to this, organizations implement federated IAM solutions along with transparent role-based structures that work in all environments.

Federated identity providers (Okta or Azure AD Federation Services) can be used to set up a shared identity strategy, so that an admin can sync roles and apply a common approach toward policy enforcement. IAM policy validation and auditing via automation can further assist in identifying deviations from best practice, ensuring that permissions remain aligned with security requirements and zero trust principles.

Challenges in Standardizing IAM Policies

IAM is another area that is common to different cloud providers but is implemented differently, creating inconsistent IAM configurations and possible attack vectors when moving resources across hybrid and multicloud environments. AWS IAM roles, for instance, are completely different beasts compared to Azure Active Directory or Google Cloud IAM, both in nomenclature and behavior. It becomes hard to keep a unified least-privilege model, which increases the likelihood of misconfiguration or an overprivileged account.

The solution is that organizations need to implement a federated IAM system that provides a single point of identity integration that can be leveraged seamlessly across the cloud platforms in the organization. This way, administrators can apply uniform access policies and keep user role synchronization throughout the environments by utilizing federated identity providers. In addition, validating IAM policy and auditing automation tools can catch any drift from best practices; they will ensure that permissions comply with security needs and zero trust architectural strategy.

Encryption Challenges and Solutions

Encryption is also one of the pillars of security at this layer, encrypting data in transit and at rest. Still, this can impact performances, especially for applications that need low latency and nontrivial processing times. Bulk encryption of key-based or large data at once, particularly in a multitenant system, can create latency issues that affect user experience as well as application performance. Organizations can take the following steps to solve this issue:

1. **Utilize Hardware Acceleration:** Several cloud providers have a hardware-based encryption accelerator—for instance, AWS Nitro Enclaves or Intel SGX—that can be used to reduce the performance overhead introduced by cryptographic operations.

2. **Use Selective Encryption Techniques:** All the data cannot be protected with the same level of safety. For sensitive or high-risk data, using encryption enables compensating controls for the less sensitive information, such as higher level segmentation within the network.

3. **Use Managed Key Services:** If possible, use services like AWS KMS or GCP KMS to minimize the operational overhead around encryption since keys are rotated and managed by the vendors and do not add much extra management burden on application workloads.

Clusters

Kubernetes, for example, facilitates application-driven clusters that serve as the workhorses of cloud-native apps. Protected clusters require some type of access and communication granulation, beginning with role-based access control. The fundamental access control mechanism that supports least-privilege access in Kubernetes is the use of RBAC policies with roles and role bindings. Developers may have read-only permissions in production namespaces but more permissive access in development clusters, minimizing the risk of unwanted or nefarious changes to production systems.

Example 16-1 shows how to use Kubernetes RBAC. Kubernetes RBAC policies enforce least-privilege principles by defining roles with specific permissions and binding them to users or service accounts. In this scenario, developers should only have access to view pods in a specific namespace without the ability to modify or delete them.

Example 16-1 *Defining and Assigning a Read-Only Role for Pods in Kubernetes*

```
# Role: Defines permissions
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
 namespace: dev-namespace
 name: pod-reader
rules:
- apiGroups: [""] # Core API group
 resources: ["pods"]
 verbs: ["get", "list"]
# RoleBinding: Assigns the role to a user
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
 namespace: dev-namespace
 name: read-pods-binding
subjects:
- kind: User
 name: developer1 # Name of the user
 apiGroup: rbac.authorization.k8s.io
roleRef:
 kind: Role
 name: pod-reader
 apiGroup: rbac.authorization.k8s.io
```

For more information, check the official documentation of Kubernetes (refer to the "References" section at the end of this chapter).

Additionally, modern tools such as External Secrets Operator or Open Bano can ensure efficient secret management practices. These tools integrate with external secret management systems (such as AWS Secrets Manager, HashiCorp Vault, or Google Secret Manager) to securely manage sensitive credentials throughout the Software Development Life Cycle (SDLC).

Example 16-2 shows how to use the External Secrets Operator, which is an open-source Kubernetes operator that retrieves secrets from external secret management tools and dynamically injects them into Kubernetes pods. This ensures sensitive data like passwords and API keys are never hard-coded or exposed in application configurations.

Example 16-2 *Retrieving and Injecting Secrets with External Secrets Operator in Kubernetes*

```
apiVersion: external-secrets.io/v1beta1
kind: ExternalSecret
metadata:
 name: db-credentials
spec:
 secretStoreRef:
  name: aws-secrets-manager
  kind: SecretStore
```

```
target:
 name: db-secret
data:
- secretKey: db-password
 remoteRef:
  key: prod/db
  property: password
```

This example allows seamless rotation and secure propagation of credentials to application pods without manual intervention.

Containers

The Containers layer has its own set of challenges due to the wide array of ephemeral environments and the isolated nature of containerized environments. The foundation of container security is validating container images for integrity and compliance because we cannot allow vulnerabilities to spread into a running container. Integration of runtime monitoring with orchestration platforms in Kubernetes can help recognize security anomalies, such as unauthorized or system calls, privilege escalation, or attempted normal operations by non-normal container processes, when the containers are live. For instance, runtime monitoring can be performed using DaemonSets to monitor all the nodes in a cluster.

Giving isolation to their containers is another major factor. Containment must ensure that containers cannot reach resources that lie outside their boundaries, and network segmentation should enforce that containers only communicate with each other as necessary, to reduce the attack surface and the impact of a compromise.

Falco is an open-source runtime security tool that plugs itself into this layer to provide an additional monitoring and threat detection mechanism. With Falco, organizations can establish and enforce security rules that identify suspicious activity in their containerized environments, which can include abnormal network connections, unexpected file access, or unauthorized privilege escalations. Falco acts like a DaemonSet and gives actionable and real-time alerts that allow Kubernetes clusters to stay secure. It is even integrated so that practitioners can take action as threats emerge and containers start to become compromised.

Code

At the foundation of cloud-native applications lies the Code layer, where security begins during the earliest stages of the Software Development Life Cycle. Secure coding practices, embedded from the start, ensure vulnerabilities are addressed proactively rather than reactively.

Example 16-3 shows an example of Input Validation, which is a critical process in this layer, serving as a frontline defense against security vulnerabilities. By ensuring that user-entered data meets specific criteria and aligns with the application's expectations, input validation prevents invalid or malicious input, such as SQL injections or improper character sets, from compromising systems. This practice not only maintains data integrity but also ensures that only properly formatted data is processed, mitigating the risk of attacks and operational errors.

Example 16-3 *Implementing Input Validation to Prevent Malformed Data and Security Risks*

```
import re
def validate_email(email):
  # Regular expression to validate email format
  pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
  if not re.match(pattern, email):
    raise ValueError("Invalid email address")
  return email
# Usage
user_email = input("Enter your email: ")
try:
  validated_email = validate_email(user_email)
  print(f"Validated email: {validated_email}")
```

```
except ValueError as e:
  print(e)
```

This example ensures that only properly formatted email addresses are processed, reducing the risk of injection attacks.

Example 16-4 shows secure password storage, which is another essential element of the SDLC—protecting sensitive user credentials even if the underlying storage system is compromised. Passwords should never be stored in plaintext; instead, they must be hashed using a strong, modern hashing algorithm—as an example, using bcrypt or scrypt to create hashes, which also ensures that user data is safeguarded in the event of a breach:

Example 16-4 *Secure Password Storage Secure Password Storage and Verification Using Bcrypt Hashing*

```
import bcrypt
# Hash a password
def hash_password(password):
  salt = bcrypt.gensalt()
  hashed = bcrypt.hashpw(password.encode(), salt)
  return hashed
# Verify a password
def verify_password(stored_hash, password_attempt):
  return bcrypt.checkpw(password_attempt.encode(), stored_hash)
# Example usage
stored_password = hash_password("SecurePassword123")
print(f"Stored Hash: {stored_password}")
if verify_password(stored_password, "SecurePassword123"):
  print("Password verified!")
else:
  print("Invalid password.")
```

Role of Cloud-Native Application Protection Platform (CNAPP)

A cloud-native application protection platform (CNAPP) brings together security functions across the application lifecycle in a single solution. Instead of stitched together capabilities like vulnerability management, runtime protection, and policy enforcement dispersed over a large landmass of disconnected tools, CNAPPs seek to converge these capabilities into a single platform. This platform improves visibility, streamlines workflows, and simplifies security complexity in fast-moving cloud-native environments, to this interaction.

Conventional security solutions included several distinct tools, each handling a given layer of the stack. One may be for scanning container images, another may be for runtime protection, and still another may be for policy enforcement. Using these scattered methods results in a higher operational overhead, creates visibility gaps, and adds a complexity to incident response. CNAPPs, on the other hand, offer a total solution for security, providing a single end-to-end security solution that removes the silos between areas of security of the application development and ensures security enforcement can be consistent.

In CNAPP, runtime protection monitors application behavior at runtime and looks for deviations from established baselines. The platform’s correlation of findings from other lifecycles, including vulnerabilities discovered in pre-runtime scans or misconfigurations in Kubernetes, further enhances this capability.

Within the application security realm, CNAPPs stand out by uniquely tackling cloud-native architecture risks like API threats, container misconfigurations, and dynamic-scale environments. They are valuable because they can make adapting to the modern application’s complexity easy while also lightening the burden of management of the security practice across multiple environments of the stack.

Choosing the right CNAPP requires careful consideration of the organization’s cloud-native footprint, complexity, and security needs. The following sections explain the main criteria that help in making this decision.

API Security

Since APIs are an essential communication component between microservices, they are a target for attackers. Threats include broken object-level authorization and excessive data exposure. The OWASP API Top 10 illustrates the risk associated with unprotected APIs. For more information, refer to the "OWASP API Security Project" (see the "References" section at the end of this chapter).

An effective CNAPP tackles such risks through capabilities such as anomaly detection, access control, and real-time detection and mitigation of API-based attacks. For example, it may identify anomalous data request patterns or block unauthorized access by authenticating token-based authentication.

Vulnerability Management

Vulnerability management helps organizations prevent zero-day exploits or other vulnerabilities that stem from misconfigurations or weaknesses in container images or Kubernetes clusters, necessitating constant scanning of cloud-native environments. CNAPPs simplify this process by automating vulnerability discovery and remediation recommendations. Being proactive about such things ensures that these weaknesses are remedied or mitigated before they can be used against the organizational resources.

Runtime Protection

Production environment applications are susceptible to sophisticated attacks that take advantage of runtime behaviors. CNAPPs use runtime security controls that watch applications, system calls, and network activity to ensure they are within defined baselines. For example, if it detects discrepancies caused by somebody trying to escalate privileges or between file access, a CNAPP may alert or even block malicious activity. To maintain security even in ephemeral environments, integrations with

Kubernetes (such as DaemonSets for node-level monitoring) provide additional visibility at runtime.

Policy Enforcement

Effective security requires consistency across infrastructure and applications. CNAPPs come with integrated policy engines that foster security enforcement at each layer. For instance, they are able to prevent the deployment of all but signed and verified container images or mandate strict resource limits in Kubernetes namespaces. In complex, multicloud environments, organizations can ensure a consistent posture and enforce their desired state with tools such as Kyverno or Open Policy Agent (OPA).

Compliance Management

CNAPPs need to automate compliance checks for industries with stringent regulatory requirements, such as healthcare and finance. They offer live visibility into configurations, compliance with policies, and possible violations to help keep the environment compliant with regulations like General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or Payment Card Industry Data Security Standard (PCI DSS). Automated reporting and auditing remove the need for manual effort to maintain compliance in a cloud-native environment.

Building Secure Applications with Cloud-Native Security

Application developers and security architects must bake security into every step of the cloud-native app lifecycle. This methodology, called *shift-left*, involves installing security in the development pipeline early to get rid of vulnerabilities instead of fixing it reactively during production. Cloud-native deployments are often made up of many microservices, containers, and APIs all working together to deliver applications with greater agility and flexibility than ever before.

Security in Application Design and Development

With cloud-native application development, the security nature is to build systems secure by design. Organizations can decrease risks in their software lifecycle by harnessing well-established security principles and integrating advanced tools and practices.

Principle of Least Privilege

The principle of least privilege means the minimum necessary permissions that an application, service, or user should have in order to perform their duties. As microservices communicate through APIs in a cloud-native realm, it is important that the API permissions are properly handled together with service roles and access controls. This minimizes the attack surface of potential compromise as much as possible by limiting access to only certain specific actions.

API Permission

APIs should be gated so that only authenticated and authorized users or services can access sensitive operations. This can be accomplished with standards such as OAuth 2.0, which allows for secure delegated access, or by implementing API gateways to aggregate authentication and authorization rules. Gateways also offer an additional layer of control on API access through traffic monitoring, rate limiting, and threat detection.

Service Roles

Organizations need to define microservices-specific roles with minimum privileges such that a service can be allowed to access what it needs to. These cluster-level limited-access controls can be defined and enforced using tools such as Kubernetes role-based access control or service meshes as Istio.

Using these tools and approaches, organizations can improve the security of microservice communication and reduce operations to those that are just authorized, realizing their principle of a trusting access strategy of least privilege.

Immutable Infrastructure

Containers and serverless functions can be redeployed automatically, increasing consistency, security, and scalability even for cloud-active environments.

Containers and Serverless Functions

Secure-by-design principles support immutability, meaning that once deployed, changes to infrastructure happen only through redeployment. This approach establishes a stable foundation for a secure ecosystem. For example, container orchestration tools such as Kubernetes can automate redeploying containers using rolling updates or blue-green deployments. These mechanisms guarantee updated versions are deployed gradually or simultaneously with almost no downtime, if at all, and the risks linked with manual changes are reduced. Similar to serverless where platforms like AWS Lambda or Google Cloud Functions automatically replace old, deployed versions with new, deployment updates are handled automatically.

Aligned with Deployment

Because an immutable infrastructure deploys from the same base image or configuration template, it prevents configuration drift. Immutable machine image creation can be achieved through tools like HashiCorp Packer, whereas IaC solutions such as Terraform or Pulumi will make sure that deployment settings are consistent across all of your settings, be it local, development, quality assurance, or production. The policies defined and embedded within these configurations remain intact during each stage of deployment, thus minimizing the risks of misconfigurations or deviations.

Secrets Management

Sensitive information, like API keys, passwords, or certificates, need to be securely stored and managed in cloud-native environments. For storing sensitive data securely, secrets management tools such as HashiCorp Vault, AWS Secrets Manager, or Kubernetes Secrets provide a secure storage mechanism, along with features like automatic rotation and fine-grained access control at each level.

- **Encryption:** Secrets should be encrypted both at rest and in transit.
- **Access Control:** Use role-based access control to ensure that only authorized services or users can access sensitive information.
- **Automated Rotation:** Implement automatic key rotation policies to minimize the risk of secrets being compromised.

Secure Coding Practices

It is the developer's responsibility to avoid broad attack vectors like injection, cross-site scripting (XSS), and insecure deserialization—all of which could be avoided with secure coding practices. Static application security testing (SAST) and dynamic application security testing (DAST) are security tools integrated into the development process to catch vulnerabilities at the earliest point in time, minimizing risks ahead of code being deployed.

These concepts will be discussed in Chapter 18, "Using Common Policy to Enforce Security," which expands on the integration of automated testing tools and policy enforcement in the SDLC.

Static Application Security Testing (SAST)

SAST tools include SonarQube or Checkmarx, which analyze source code as part of the development process to identify vulnerabilities like SQL injection or a failure to properly validate input. They seamlessly integrate into CI/CD pipelines to scan the code on every commit or pull request. Tools like SonarQube give automated feedback in the developer IDE or in the build process and even before the code goes into production, flagging the defects. With SAST tools embedded in the pipeline, organizations can adopt a proactive approach toward secure coding, ensuring that vulnerabilities are addressed early in the SDLC.

Dynamic Application Security Testing (DAST)

DAST tools try to emulate real-world attacks on running applications to find vulnerabilities in runtime environments; one example is OWASP ZAP. ZAP can check an endpoint for injection vulnerabilities or misconfigured headers and give developers actionable feedback to fix it. Adding DAST in further stages of the SDLC enables developers to detect and remediate the vulnerabilities that a static code scan cannot see, prior to deploying the application.

CNCF Tools for Secure Coding

Some CNCF tools can also help in assessing and alleviating security vulnerabilities in the development pipeline:

- **KubeLinter**: A linter (a static analysis tool) for Kubernetes manifests that checks for security best practices and common misconfigurations, helping you write and maintain secure Kubernetes manifests.
- **Trivy:** A container image and IaC template vulnerability scanner for GitHub and GitLab, easily providing early feedback in the development lifecycle.
- **Tekton Chains:** A tool that adds an extra layer of security to your CI/CD pipelines by enriching the builds with provenance and protecting them with cryptographically signed metadata.

By using these tools together, the organization can build a proactive and secure coding framework across the SDLC that results in improved application security and compliance with organization policies.

Shift-Left Security and DevSecOps Integration

Shift-left security and DevSecOps are really two sides of the same coin that collectively help secure cloud-native applications. Integrating security in the early stages of development, as well as CI/CD (CI/CD drives continuous development and deployment), organizations can work on preventing any vulnerabilities and security risks before they end up as major issues.

DevSecOps: Embedding Security into DevOps

DevSecOps, as an evolution of DevOps, spreads security across every phase of the development lifecycle. Security is not merely an afterthought, but apart as an ingredient in processes from design to deployment where security is now verified continuously and in an automated manner.

- **Security Automation:** CI/CD pipelines are able to integrate scanning tools to automatically validate code and configurations as new code is committed.
- **Ongoing Security Testing:** Security tests within the pipeline notify developers of any issues at an early stage, making resolutions cheaper and easier than in subsequent development or production cycles.

Top Cloud Security Risks in DevOps

DevOps significantly improves productivity, but it also introduces its own security challenges:

- **Misconfigurations During Runtime:** Unlike traditional misconfiguration omnipresence in other areas, most of the runtime misconfigurations do not come to light in the earlier stages.
- **Developer-Security Misalignment:** Developers and security teams often lack common visibility or context in regard to security problems in the early stages of development, which creates miscommunications.
- **Dynamic Nature of the Cloud Environments:** The dynamic nature of the cloud infrastructure makes the base challenge of maintaining security configuration adherence and compliance.

It is imperative to have security as a part of the DevOps workflows, starting from ensuring that both application code and infrastructure underpinning are secure.

Strategies for Enhancing Collaboration

DevSecOps success is achieved when developers, security teams, and operations work together. Strategies include

- **Shared Dashboards:** Dashboards with security metrics and findings within the dashboard keep teams transparency and alignment in real time.

- **Security Champions:** Having security champions in the development teams helps in embedding the security in the development process. These people are the bridge between security and development teams.
- **Integrated Feedback Loops:** CI/CD pipelines include feedback loops that allow developers to correct errors quickly while remaining agile.

Managing Dynamic Cloud Configurations

Dynamic cloud environments require solutions that can adapt to constant changes while ensuring security and compliance. Beyond Infrastructure as Code, organizations can leverage complementary tools and practices to enhance configuration management and reduce risks.

Configuration Management Tools

Configuration management tools provide centralized storage for sensitive parameters such as API keys, database credentials, and application settings, ensuring secure and consistent deployments. These tools prevent misconfigurations by integrating with deployment pipelines and runtime environments. Examples include AWS Systems Manager Parameter Store and Azure Key Vault for cloud-native environments, as well as the CNCF projects that help utilize secure registries to hold configuration data to be used as part of GitOps pipelines, and systems to inject parameters dynamically into Kubernetes clusters.

Continuous Monitoring and Drift Detection

Real-time monitoring and drift detection tools ensure that cloud configurations remain consistent with predefined baselines by identifying unauthorized changes or deviations. These solutions often provide automated alerts and remediation workflows to minimize downtime and risk. Examples include Cloud Custodian, a governance tool that automates policy enforcement, and Driftctl, which focuses on detecting and resolving configuration drift in cloud infrastructure.

Dynamic Secrets Management

Dynamic secrets management solutions secure sensitive information like credentials, tokens, and encryption keys by rotating them regularly and limiting their lifespan. These tools integrate with cloud-native platforms to deliver secrets dynamically at runtime. Examples include HashiCorp Vault, a widely used tool for secrets lifecycle management, and External Secrets Operator, which connects Kubernetes clusters to external secrets management systems for runtime injection.

Automated Remediation Tools

Automated remediation tools streamline the resolution of configuration issues by triggering predefined workflows in response to policy violations or security risks. These solutions help enforce compliance and mitigate misconfigurations in real time. Examples include Amazon Config Rules and Google Cloud Config Validator for automating compliance checks, alongside CNCF project Kyverno, which enables policy-based remediation directly within Kubernetes environments. Another approach for predefined execution can be done using runbooks and playbooks. Tools like PagerDuty Runbook Automation help teams respond to configuration changes by automating predefined workflows for resolving issues. These types of automation logics can also be embedded directly into **Kubernetes controllers and operators**, enabling more self-autonomy by automatically responding to configuration changes or state events within the cluster.

Infrastructure as Code (IaC) and Security

Infrastructure as Code is a new way to manage and automate your infrastructure by writing code rather than using manual processes that you have traditionally kept for deployments. Terraform, AWS CloudFormation, and Azure Resource Manager have the ability to provision infrastructure but also put security policies into automation so they will not be missed.

To implement secure and compliant IaC and boost security, organizations can include IaC checks throughout the CI/CD pipeline. This ensures that potential security vulnerabilities are found before they make it to production and keeps infrastructure configurations consistent.

- **Static and Dynamic IaC Analysis:** Static analysis checks code for security vulnerabilities before deployment, while dynamic analysis tests the infrastructure in real-world conditions. Tools like Checkov or TFSec scan IaC templates for misconfigurations or noncompliance with security policies.

- **Developer Tool Integration:** When security checks are integrated into developers' existing tools, such as code editors or CI/CD platforms, security becomes part of the workflow, reducing friction and ensuring compliance without disrupting productivity.

- **Cloud environment change policy:** The cloud infrastructure will keep evolving, which makes it difficult to maintain security configurations and compliance with changes in the dynamic nature of that environment.

Securing the Software Supply Chain

In a cloud-native development environment, you have to think about the utilized software supply chain's security at the first place. Whether it be first-party code or third-party dependencies, a secure supply chain ensures that everything is secure, compliant, and running the latest stable and vetted versions. Everything from the source code to the build process to the container image and the runtime dependencies make up the software supply chain, which makes it a key area of application security.

Understanding the Function of a Software Bill of Materials (SBOM)

Some SBOMs allow you to see what components and dependencies your apps may contain. This encompasses libraries, modules, container images, and provides a more transparent view of all software components. An SBOM serves to log these various parts to enable organizations to

- **Track Vulnerabilities:** Identify and mark obsolete or insecure dependencies, such that they can be quickly patched once security vulnerabilities are detected.

- **Stay Compliant:** Adhere to any industry regulations that require visibility into software composition such as NIST standards or Executive Order 14028 on cybersecurity.

Container scanning tools like Anchore and Clair, for instance, scrutinize images and their dependencies to deliver rich SBOMs, exposing vulnerabilities at every layer of the stack.

Securing Trusted Components in the Software Supply Chain

By trusting components, you reduce the risk of adding vulnerabilities to production. Trusted components are artifacts that are signed with cryptographic proof of integrity and authenticity, and are always monitored (tracked for updates and vulnerabilities) and controlled, which means they only reach production after certain security policies are built in.

Examples of trusted component practices include

- **Signing Artifacts:** Tools like Notary and Sigstore make sure that, before container images and binaries are deployed, they are signed cryptographically.
- **Dependency Validation:** Tools such as Dependabot automatically scan source code repositories for insecure dependencies.
- **Container Origin:** Tekton Chains are an example of a platform that provides signed metadata about container images, ensuring both traceability and integrity.

Dependency Tracking and Tracking Validation

Dependencies are among the biggest attack vectors in the software supply chain. To ensure that dependencies **remain safe** against vulnerabilities throughout the entire application lifecycle, organizations must track them. Some key practices to follow for managing dependency security include

- **Real-Time Scanning:** Tools such as Trivy or Grype actually scan container images, dependency trees, and IaC templates for vulnerabilities on the fly.

- **Policy Enforcement:** Builds with unsupported or vulnerable dependencies that do not meet corporate security policies are automatically denied.
- **Scale Monitoring:** Clair is an open-source project that does static analysis on container images where vulnerabilities are tracked in real time.

Component Integrity: Building Trust

Protecting the software supply chain from maliciously altered components intended for production requires integrity assurance. A big part of that is implementing tight control over what goes into both the development and runtime environments. Some strategies for enforcing component integrity might be to use

- **Verifiable Images:** Utilize tools like Harbor, which are trusted registries that ensure image signing and vulnerability scanning before deployment.
- **Zero Trust for Artifacts:** Use only signed, validated artifacts and runtime verification to detect tampering.
- **Automated Build Pipelines:** Integrate tools like Jenkins or GitLab CI equipped with security checks to validate each artifact against a defined security baseline.

API Security

APIs serve as the spine of cloud-native applications by allowing seamless communication between the microservices. But that makes APIs a major attack vector as well, due to this dependency on them. API security is essential to ensure the data communication between the services remains intact, confidential, and available. The best practices outlined in the following sections are essential for API security.

Authenticating and Authorizing

Robust authentication and authorization methods make sure that only services and users that are supposed to access the APIs are able to do so.

- **Protocol:** OAuth 2.0 and JSON Web Token (JWT) are compact, URL-safe means of representing claims to be transferred between two parties, but most importantly they are cryptographically-cannot-be-changed tokens (well, mostly).
- **RBAC:** Role-based access control enables you to define more granular permissions and capabilities for your users and services.

Rate Limiting and Throttling

Organizations can limit the number of API requests that one user or service can execute in a time span to prevent abuse. Such controls are especially helpful for preventing denial-of-service (DoS) attacks. Rate limiting policies can be applied in API gateways either by token bucket or leaky bucket algorithms to maintain fairness and prevent overload.

Encryption

Encrypting API traffic protects sensitive data from interception or tampering during transit and ensures secure storage at rest.

- **In Transit:** Encrypt API communication channels with TLS 1.2+ and keep data in an encrypted state when being transmitted.
- **At Rest:** Wherever possible, ensure sensitive data is buried under AES-256 and simply encrypt it at rest.

API Gateway

API gateways provides centralized control of API security and performance. They serve as gateways between the client applications and the backend services, enforcing critical security points.

- **Role:** API gateways such as Kong, NGINX, or AWS API Gateway are used for rate limiting, authentication, authorization, or monitoring of traffic.
- **Encryption Management:** Gateways are aware of HTTP traffic, so they are able to encrypt all the API traffic over HTTPS and manage your TLS certificates on the backend services.

Secure APIs Using Kubernetes Security

At its core, Kubernetes is an API-driven platform, and it goes without saying that protecting the APIs is key to protecting the control plane, the workloads, and everything else that makes up a Kubernetes cluster. As already highlighted in greater detail, a lot of Kubernetes security practices, by nature, protect APIs, including aspects such as audit logging, secrets management, and supply chain security.

The API-driven nature of Kubernetes makes it inherently suited for implementation of zero trust. With zero trust, no one—user, service, or device—is trusted by default; instead, every interaction must be continuously validated and access enforced according to the principle of least privilege.

Within the realm of API security, these are uniquely protective practices: Audit logs help perform anomaly detection from a finer granularity on top of API activity, secrets management protects the sensitive credentials used in API calls, and supply chain security prevents unwanted components from communicating with Kubernetes APIs. Collectively, these measures remove API attack surface, helping Kubernetes clusters stay safe from access and use by unauthorized parties.

Now let’s review some areas that require special focus to withstand the specific nature of APIs in a cloud-native context, with Kubernetes.

Role-Based Access Control (RBAC) and API Security Controls

The Kubernetes API server is the core component of the cluster control plane, and it can be the single biggest attack vector. By hardening this component, you will make sure only the authorized users and services are able to communicate with Kubernetes objects.

- **Fine-Grained Permissions:** Create roles with exact permissions using role-based access control and attach those roles to service accounts, users, or groups on a least privilege basis.
- **Periodic Audits:** Continuous validation of trust is aided with regular audits of API logs for unauthorized access attempts.

- **API Server Hardening:** Use secure default features in the API server, such as authentication, transport layer encryption via TLS, and more fine-grained access to specific endpoints.

Network Policies

Network policies are the Kubernetes equivalent of firewalls; they govern how pods and other services are allowed to communicate. In addition to preventing unauthorized entities from being able to call the API, these policies are also important to prevent lateral movement and make sure that the scopes are being enforced.

- **Granular Traffic Management:** Set up network policies to ensure only trusted pods and services can access internal APIs.
- **Default-Deny Policies:** A default-deny network policy prevents all traffic by default (need explicit rules to allow traffic), which is the recommended way to go with the policies.
- **Better Security Tools:** Solutions such as Calico or Cilium can enforce DNS-sensitive policies, include mDNS between two pods, and securely encrypt API communication to the pods.

Container Isolation

Kubernetes workloads do proper container isolation as an extra security implementation, because internal APIs do facilitate communication between the microservices running across nodes and within the cluster. Thus, enforcing strong isolation helps prevent unauthorized interactions and limits the lateral movement in case a component is compromised, which aligns with zero trust security principles.

Kubernetes provides different security mechanisms like admission controllers to enforce rules at deployment time, while third-party tools such as Open Policy Agent (OPA) allow dynamic policy enforcement across Kubernetes resources. It’s important to clarify that Kubernetes does not provide full container isolation by default; it will require additional security configurations such as Pod Security Admission (PSA), NetworkPolicies, and RuntimeClass.

For further kernel-level isolation, sandboxing technologies like gVisor and Kata Containers could enhance workload security by restricting the direct access to the host kernel, which helps mitigate container escape risks and eventually limits the attack surface. These solutions can be integrated as part of a defense-in-depth Kubernetes security strategy.

Resource Limiting

Zero trust extends beyond authentication and authorization to include resiliency against resource abuse. Kubernetes resource limits and quotas enforce fair usage policies for API that can potentially be abused, whether intentionally or accidentally.

For instance, APIs are susceptible to resource exhaustion attacks like high server load by a huge number of API calls or unoptimized workloads and overwhelming API server. Good resource management avoids degradation of service.

- **Quotas and Limits:** Set user and name-space level CPU, memory, and API call quotas and limits.
- **Autoscaling:** Express Horizontal Pod Autoscalers (HPA) and Cluster Autoscalers settings to avoid unnecessarily tight usage of the specific API endpoints.

Further Thoughts on Kubernetes API Security

Though the practices mentioned in the preceding sections focus on some critical dimensions of API security in Kubernetes, other considerations can contribute to your defense-in-depth:

- **Ingress and Egress Control:** Ingress controllers are key to securing the external exposure of the API, ensuring authentication, rate limiting, IP whitelisting, and so on. Egress control helps you to get pods to talk only with trusted outside services.
- **API Observability:** Use monitoring tools such as Prometheus and Grafana to visualize API call patterns, bring up the unusual behavior of the APIs, and determine whether any abuses are statistically possible on APIs.

- **Integrate Service Mesh:** Use service meshes like Istio or Linkerd to enable mutual TLS (mTLS) for securing and controlling API communication and to enforce fine-grained access control policies.
- **Secrets Management:** Protect sensitive API keys and tokens using Kubernetes Secrets and enrich them with external tools like HashiCorp Vault/Open Bao or Secrets Operators (see the previous section) for dynamic rotation and access control.

Unique Security Considerations for Serverless Architectures

Serverless computing represents a shift in cloud-native architecture, offering scalability, ease of deployment, and operational efficiency. However, these advantages come with unique security challenges. In traditional architectures, much of the security focus revolves around managing infrastructure and maintaining the perimeter. With serverless computing, the perimeter dissolves, requiring security to move closer to the application and function levels. Common serverless platforms include AWS Lambda, Azure Functions, and Google Cloud Functions, which abstract the infrastructure layer, leaving developers to focus on application logic.

Additionally, CNCF-backed projects such as Knative and OpenFaaS provide open-source alternatives, enabling organizations to deploy and manage serverless functions in Kubernetes environments. These tools align with cloud-native principles while allowing more control over deployment environments compared to commercial offerings.

Serverless Shared Responsibility Model

Serverless security responsibilities are split between provider and application owner. Providers take care of infrastructure security at the level of data centers, physical and virtual networks, servers, and operating systems, while all critical aspects of application ownership will remain with the application owners. These also consist of secure application logic, code quality, data protection and configuration security. The challenge is new because this division takes away some of the control organizations traditionally relied on with layers of security built around the perimeter with firewalls and host-based protections, and instead, it means the applications now need to secure against a much larger attack surface across many more event sources and cloud services. Success comes from knowing these perimeters and putting compensating controls and cabinets on the application layer. Figure 16-1 shows an adjusted view of the Shared Responsibility Model from the perspective of serverless ownership.

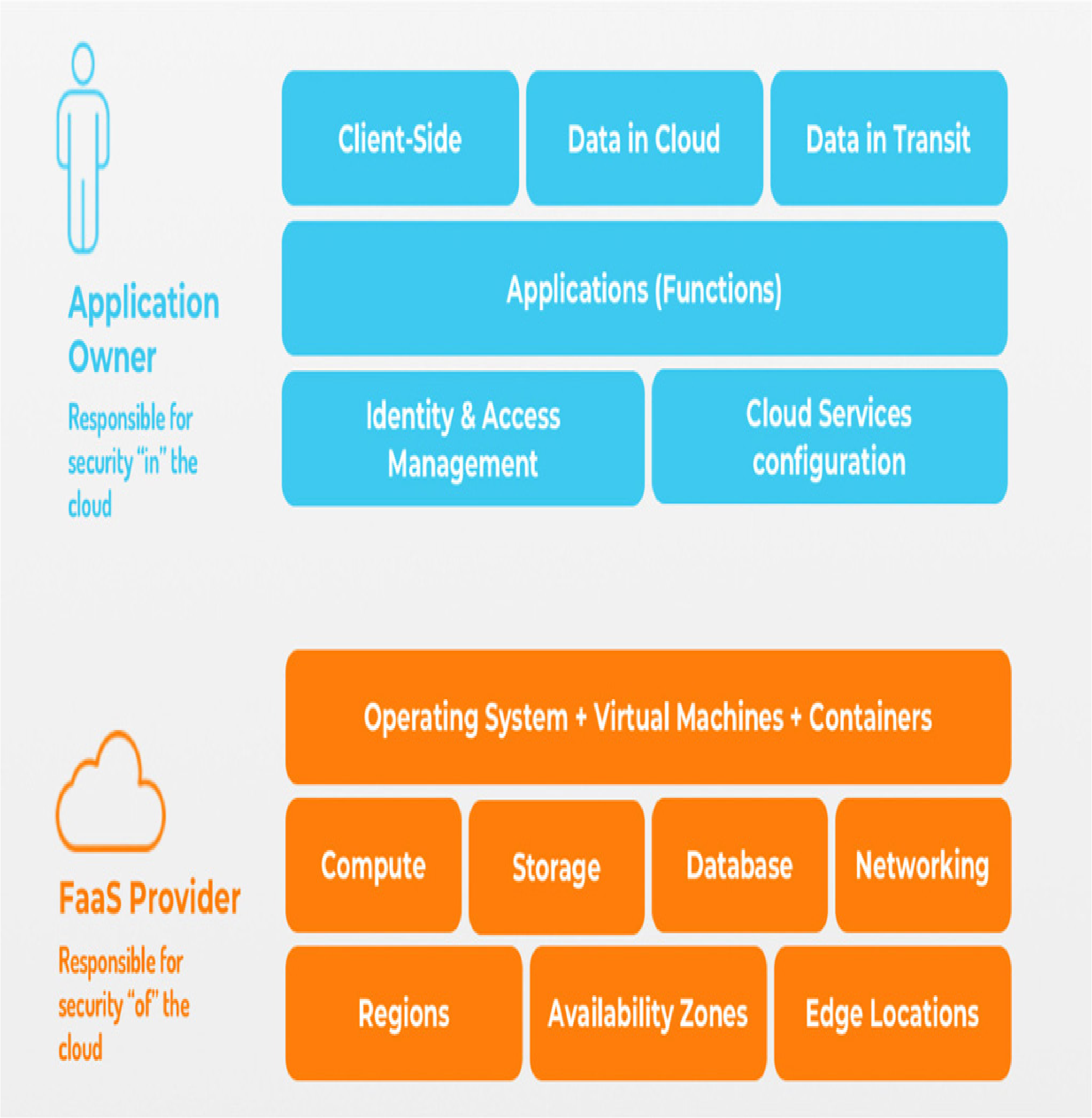

Application Owner
Responsible for security "in" the cloud
Client-Side
Data in Cloud
Data in Transit
Applications (Functions)
Identity & Access Management
Cloud Services configuration
FaaS Provider
Responsible for security "of" the cloud
Operating System + Virtual Machines + Containers
Compute
Storage
Database
Networking
Regions
Availability Zones
Edge Locations

Figure 16-1 *The Shared Security Responsibilities Model, Adapted to Serverless Architectures (Source: https://github.com/puresec/sas-top-10)*

Key Serverless Security Challenges

The serverless attack surface is different, focusing on code-specific security, granular function permissions, and API security. Serverless functions are inherently dynamic and ephemeral; hence, their security model has to be aligned to these characteristics:

Function-Level Permissions

All the functions must work with the least privilege, following the least-privilege principle. Incorrect permissions can be exploited for unauthorized access or lateral movement in the environment. It becomes unscalable to manage IAM roles manually, as the number of serverless functions increases.

Dependency Management

Many third-party libraries are used with the serverless function. If these dependencies are not current or vulnerable, they can create risk in the environment. While these threats are serious, automating scanning for dependencies during development and periodic audits can alleviate much of the risk.

API Gateway Security

APIs are usually the entry points to a serverless application; thus, it is one of the most attacked serverless components because they get exposed to risks like API scraping, injection, or denial of service.

Visibility and Monitoring

Due to their ephemeral nature and inherent distributed architecture, serverless environments generate large volumes of log and metric data. Aggregating and analyzing this data are essential to gain actionable insights and ensure security across the stack. Real-time monitoring tools are key for finding anomalies when it comes to the rate of the traffic, function execution, or access attempt.

Proactive Threat Detection and Response

In serverless architectures, traditional monitoring techniques may lack visibility due to the abstraction of the underlying infrastructure; proactive threat detection and response become critical. Thus, organizations should aim to use leveraging tools and services that not only detect threats but also respond to them proactively in real time. Attacks can be detected and mitigated with the help of behavior anomaly detection, dynamic alerting systems, and runtime monitoring tools.

Data Leakage

Serverless functions store loads of logs that may not be handled properly and result in exposing these logs containing sensitive information. Sensitive data like API keys or personally identifiable information (PII) must be excluded from log entries at the configuration level.

The Path Forward for Serverless Security Best Practices

Securing serverless environments requires organizations to adopt both tactical best practices and strategic approaches tailored to address the unique challenges of serverless architectures. The following strategies balance immediate operational measures with a forward-looking vision to ensure security at scale.

Enforce Fine-Grained Permissions

Limit each serverless function's access to only the resources it needs, adhering to the principle of least privilege. Regular audits prevent over-privileged roles, while automated solutions dynamically manage permission updates as environments evolve. Tools like SPIFFE/SPIRE and AWS IAM Access Analyzer help ensure secure workload identities with temporary credentials.

Manage Dependencies Proactively

Since serverless functions are heavily reliant on third-party libraries, operators must ensure that dependencies are maintained well and regularly updated and scanned for security issues constantly. A better approach is to build dependency audits properly into the CI/CD pipeline and identify potential risks during the development process rather than later in production. CNCF tools like Trivy could automate this process and help to achieve a safer deployment.

Harden API Gateways

API gateways are critical components in a serverless architecture, acting as an entry point for the serverless applications. Therefore, it is important to enforce good authentication and authorization mechanisms (e.g. OAuth2 or JWT), use rate limiting to deter abuse, and monitor the API traffic for abnormal patterns that are an indicator of scraping, injection, or DoS attacks. Kong Gateway and AWS API Gateway do a great job on managing security for APIs.

Enhance Visibility and Real-Time Monitoring

The ephemeral nature and distributed architecture of serverless environments generate a vast amount of log data and metric data. Anomaly detection tools (such as AWS CloudWatch and Fluentd) are essential to scan for unusual traffic patterns, unexpected functions executions, or unauthorized access attempts. Having a holistic view of all serverless logs enables organizations to quickly identify any possible vulnerabilities or breaches and react accordingly, whether it be through automated alerts or a manual response.

StratoShark is a detection solution to monitor cloud-native and serverless environments at the level of system calls, giving you details and context beyond what traditional tools can do. StratoShark goes beyond higher-level usage patterns monitored by tools such as AWS CloudWatch and Fluentd (for instance, abnormal invocation rates or execution of a function by an unauthorized caller) and syscalls (which may indicate small-scale security incidents such as privilege escalation or unauthorized resource access). The multicloud compatibility allows consistent monitoring across event-based function services such as AWS Lambda, Azure Functions, and Google Cloud Functions.

Protect Sensitive Data and Logs

Data needs to be encrypted at rest and in transit with strong encryption standards provisioned and controlled with tools such as Cloud KMS or Key Vaults; the reason is to reduce the risk of sensitive data exposure. Therefore, it is also crucial to ensure that the logging practices cleans the logs and redacts sensitive fields and enforces secure storage policies in a way that guarantees that sensitive data will not be stored in plaintext so that there are little chances of data getting leaked while debugging or troubleshooting.

Automate Security Validation and Compliance

In serverless architectures, it is impractical to perform security and configuration checks manually. However, automated security validation also ensures that configurations, deployments, and the infrastructure are compliant and that security rules have not deviated. Organizations need to integrate compliance checks with CI/CD pipelines; you integrate the checks into the CI/CD process so that you capture and prevent misconfigurations or noncompliance at development and deployment stages.

Scanning continuously during runtime for any vulnerabilities and misconfigurations adds another layer of security. It alleviates operational overhead and helps meet the compliance standard for all environments (that is, GDPR, HIPAA, and PCI DSS.).

Proactive Threat Detection and Response

Going beyond traditional monitoring means organizations need solutions and services that can actively detect and respond to incidents as they occur. Using behavioral anomaly detection, dynamic alerting systems, and runtime monitoring tools, you can identify an attack as fast as possible and take action.

Tools like AWS CloudWatch and Fluentd can track anomalies, such as too high invocation frequency, unauthorized function access, and so on, or spikes in CPU, memory, or disk usage. This allows security teams to take actionable intelligence to rapidly respond to potential breaches or misconfigurations.

Collaborative Security with DevSecOps

A DevSecOps model will enforce security practices to be integrated into the development lifecycle at the very beginning. Integrating security tools such as static application security testing (SAST) and dynamic application security testing (DAST) into CI/CD pipelines helps to detect vulnerabilities early and fix them before they become risks at deployment time.

Encouraging the cooperation of development, infrastructure and safety teams is important. Designating champions among development teams will support security in code, determine knowledge gaps, and create a smoother communication flow. As a result, it reinforces the general cybersecurity posture of serverless architectures.

Adopt a Security-First Mindset with Automation

In a serverless environment, a security-first approach starts with protection at every level of the stack. At scale, automation is key to enforcement of security principles consistently:

1. Implicit use of dynamic permission management through temporary credentials guarantees that functions will have scoped access to the resources that they need, and only those resources.
2. Enforcing encryption for sensitive data at rest and in transit protects data from exposure during breaches.
3. Policy as Code frameworks such as Open Policy Agent support scalable, fine-grained policy enforcement across infrastructure and applications.

Automation minimizes human error, guarantees adherence to security standards, and enables security to scale along with the explosive growth of serverless environments.

Critical Attack Vectors in Serverless Applications

References such as PureSec Serverless Application Security Top 10 (now archived on GitHub) and OWASP Serverless Top 10 highlight the most important serverless security risks. They focus on vulnerabilities specific to cloud deployments—for example, insecure APIs, insufficient permissions, improper dependency management, along with the implications these have on security approaches. For more information, refer to the “OWASP Serverless Top 10” and “The Ten Most Critical Risks for Serverless Applications v1.0”; see the “References” section at the end of this chapter.

OWASP ServerlessGoat is an intentionally vulnerable serverless application built to provide a wealth of knowledge for developers and security professionals to learn about these risks. It is a practical tool for investigating and mitigating the most prevalent security vulnerabilities, providing real-world experience with serverless security threats. For more information, refer to the “References” section at the end of this chapter.

Here is more of an abstract and high-level classification of serverless attack vectors. As such, these patterns show how multilayer validation, tight permission boundaries, and continuous audits and monitoring become imperative for serverless security professionals. The following sections describe several key categories of serverless security threats.

Function Input Manipulation

This type of service takes inputs from various sources, such as HTTP APIs, queues, and IoT devices, and runs code in response. Input validation vulnerability is exploited to inject malicious payloads. A function that handles JSON events might be a NoSQL injection risk if it takes no precautions when using values found in nested fields (exploiting embedded documents in NoSQL databases) against the database.

Access Control and Authentication

Authentication is complicated in the distributed world of serverless architectures. Temporary credentials or assumed roles can be used by functions, creating extra room for permission escalation. If IAM policies are overly permissive, An attacker who compromises one function could potentially access unrelated resources. The challenge is exacerbated by the complexity of securely handling secrets across multiple functions.

Resource and Configuration Attacks

The denial of wallet attack works by exploiting the serverless billing model, causing high amounts of function executions or increasing the execution duration. In addition, a broader attack surface also includes the use of open permissions and too-high timeout settings when deployed. As an example, an attacker may force simultaneous calls to a costly function, leading an organization to quickly exceed its cloud budget.

Event Injection

Most serverless applications are event-driven, which means functions are invoked in response to events from other services such as message queues, storage events, or webhooks. Functions automatically executed on triggers can be exploited by attackers to inject arbitrary events and trigger functions to execute unwanted logic. An example would be an attacker forging webhook payloads that overload a function like event schema or improperly validating the event schema to create unwanted side effects.

Cold Start Abuse

Serverless functions automatically scale and create new instances when needed. When a new instance of a function is initialized, it is known as a *cold start*. It adds latency, and in some cases, the uninitialized states are exposed, and sensitive data may be leaked. This means that attackers can purposefully trigger many cold starts to attack and degrade performance, and can identify weaknesses during initialization, such as misconfigured variables or poorly secured secrets.

Other Common Security Risks Witnessed

The OWASP Top 10 is still the ultimate reference for identifying the most common application security risks. It is based on data gathered from thousands of organizations, over 100,000 real-world applications and APIs.

This extensive data is processed and ranked according to various parameters, including (but not limited to) ease of exploitation, ease of detection, and impact of successful exploitation, and yields the top web application security risks.

For deeper exploration of serverless security vulnerabilities and best practices, reference these projects directly (see the “References” section at the end of the chapter).

Detailed Security Flow Through Components

The following sections explain step-by-step how the security mechanisms are in practice for a serverless architecture, as depicted in Figure 16-2. Requests pass through multiple components; security checks and monitoring are done at different layers to address integrity and performance issues for the application and the overall system.

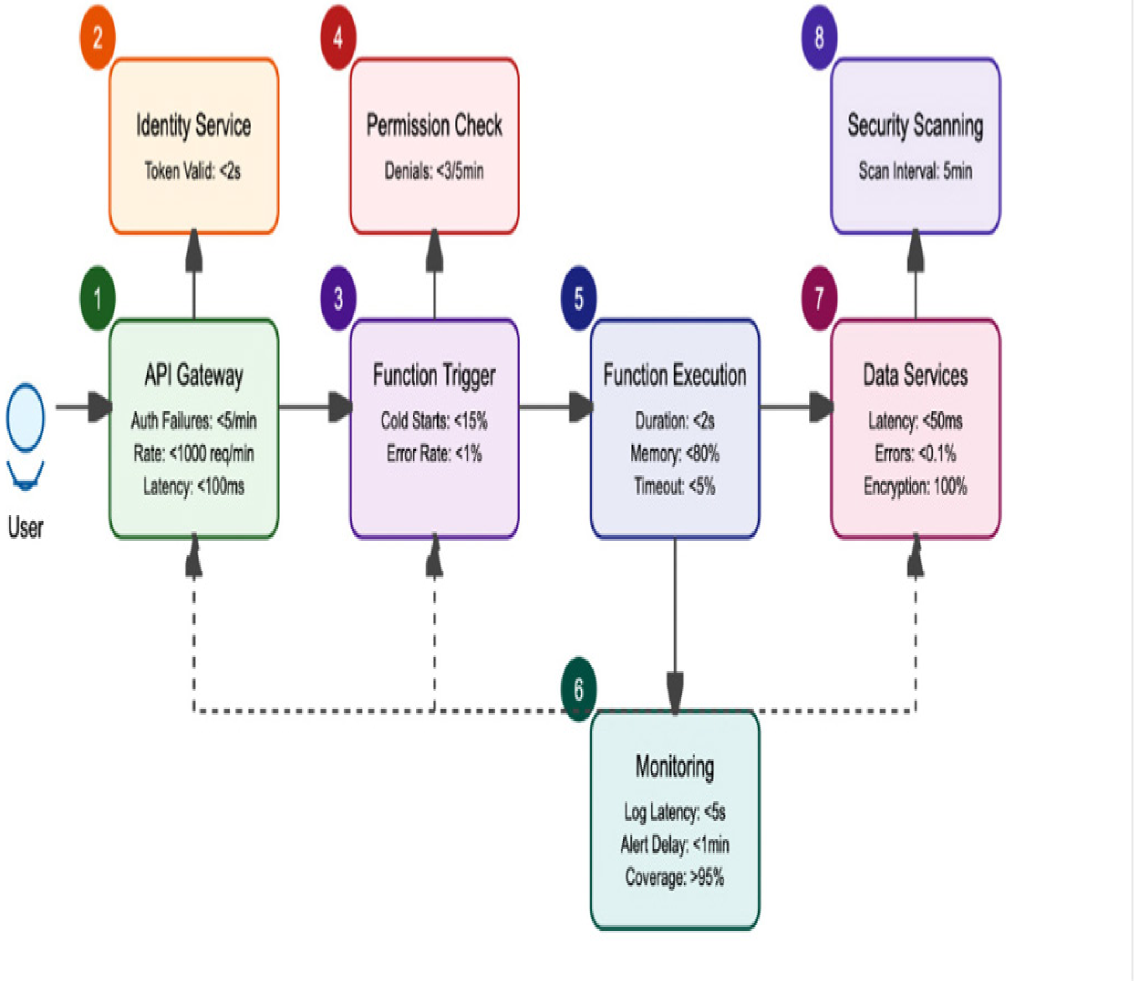


Figure 16-2 *Detailed Serverless Security Flow with Metrics*

1. API Gateway Processing

   The security flow starts when a user requests access to one of the serverless functions through the API Gateway, the first point of entry and main security checkpoint. The request contains the necessary authentication credentials (typically a JWT token) along with the desired payload. In this context, the user could represent any authenticated entity, such as an end-user app, another service, or a component of the same system needing serverless resources.

   The API Gateway carries out several essential security functions:

   Integration with Identity Service to validate the JWT token.

DDoS attempt prevention and resource abuse protection through rate limiting.

Payload verification via schema validation to ensure request integrity.

Request routing to the appropriate backend services, ensuring secure communication flow.

Metrics Monitored:

Authentication failures: <5/min

Request rate: <1000 req/min

Gateway latency: <100ms

Invalid payload rejections

Token validation success rate

2. Identity Service (Orange)

After the API Gateway forwards the request, the Identity Service validates the authentication token. This step ensures that the user or entity requesting access is authenticated before any further processing. The Identity Service verifies the token's integrity, checks for expiration, and ensures that the token was issued by a trusted authority.

The Identity Service performs the following security tasks:

Token Validation: Checks the authenticity and validity of the JWT token.

Session Management: Handles user sessions and ensures tokens are not reused maliciously.

Authorization Checks: Confirms that the token grants the appropriate level of access.

Revocation Verification: Ensures the token has not been revoked due to security incidents.

Metrics Monitored:

Token validation duration: <2s

Token expiration rate

Session validation success rate

Revoked token rejection count

Authorization failure rate

1. User Request Initiation

The security flow starts whenever a user requests to access one of the serverless functions. The request contains the necessary auth credentials (typically JWT token) with the desired payload. In this context, the user could represent any authenticated entity such as but not limited to an end-user app, another service, or a component of the same system needing serverless resources.

Metrics Monitored

- Success rate of request initiation
- Authentication token presence
- Request payload size and format validation

2. API Gateway Processing

The API Gateway is the first point of entry and main point of security. It carries out several essential security functions:

- Integration with Identity Service to validate the JWT token
- DDoS attempt and resource abuse prevention through rate limiting enforcement
- Payload verification via schema validation
- Request routing to the appropriate services

Metrics Monitored

- Authentication failures: <5/min
- Request rate: <1000 req/min
- Gateway latency: <100ms
- Invalid payload rejections
- Token validation success rate

3. Function Trigger Evaluation

The Function Trigger controls the instantiation of the serverless function, which means it

- Handles cold start scenarios
- Sets up a secure execution runtime environment
- Sets up appropriate context security for function
- Handles scaling and availability of the function

Metrics Monitored

- Cold starts: <15% of invocations
- Error rate: <1%
- Trigger latency
- Security context initialization time
- Resource provisioning success rate

4. Permission Check Verification

This part implements the least-privilege principle by ensuring

- IAM policies specific to the requested function
- Enforcement of role-based access control

- Creation of temporary scoped credentials
- Evaluation and enforcement of access policy

Metrics Monitored

- Permission denials: <3/5min
- Policy evaluation time
- Credential generation latency
- Role assumption success rate
- Policy cache hit rate

5. Function Execution Management

When the function is actually executed, the system

- Keeps execution environments isolated
- Imposes resource policy and security boundaries
- Handles the lifecycle and resources of functions
- Manages error cases and automatic retries

Metrics Monitored

- Execution duration: <2s
- Memory utilization: <80%
- Timeout occurrences: <5%
- CPU utilization
- Network usage
- Error handling success rate

6. Monitoring Layer

The monitoring system offers comprehensive observability:

- Collects logs from all components
- Produces alerts and notifications from security incidents
- Retains control over security posture of their system
- Handles cross-platform event correlation

Metrics Monitored

- Log latency: <5s
- Alert delay: <1min
- Monitoring coverage: >95%
- Event correlation accuracy
- Alert accuracy rate
- System visibility coverage

7. Data Services Integration

The Data Services layer handles all the data operations:

- Requires secure access patterns to the data
- Supports encryption for both data at rest and in transit
- Manages data access credentials
- Deals with requests for data operation

Metrics Monitored

- Operation latency: <50ms
- Error rate: <0.1%
- Encryption coverage: 100%
- Cache hit rate
- Connection pool utilization

- Data operation throughput

8. Security Scanning

Continuous security assessment occurs through

- Regular vulnerability scans
- Security posture evaluation
- Integration with monitoring systems
- Proactive threat detection

Metrics Monitored

- Scan interval: 5min
- Vulnerability detection rate
- False positive rate
- Scan coverage
- Remediation time
- Risk score trends

Database Access Example

While not explicitly shown in Figure 16-2, database access integrates with the flow through the Data Services component (Step 7):

- After successful permission verification (Step 4), the function receives a SPIFFE ID.
- Using this ID, temporary credentials are requested from AWS STS.
- These credentials are used through Data Services (Step 7) to access the database.
- Monitoring (Step 6) tracks all database operations and access patterns.

Additional Database-Specific Metrics

- Query execution time
- Connection pool utilization
- Query error rate
- Cache hit ratio
- Dead lock occurrence rate
- Replication lag (if applicable)

This means that the database access follows security principles and monitoring patterns defined in the main flow while bringing specific database-related security controls and metrics.

Emerging Trends and Future Outlook in Cloud-Native Security

APIs and microservices form the backbone of modern cloud-native architecture, enabling agility and scalability. However, they also bring unique security challenges, particularly in multicloud and hybrid environments. Organizations must turn to proactive solutions, which bring together continuous automated discovery and scanning for vulnerabilities, as well as anomaly detection, to protect such hybrid environments from cyber attacks.

Architected around a zero trust model, CNAPP solutions implements strict authentication and access controls by default, never trusting anything or anyone by default. These solutions thus provide a new level in protecting your cloud with the capability of detecting the threats and responding quickly and effectively through advanced API traffic monitoring.

Reimagining Cloud Security and Zero Trust with CNAPP Solutions

Meet your new secret weapon for cloud-native application and API security: CNAPP. It combats traditional beliefs, justifies a holistic approach to security, and offers proactive solutions to make businesses future-proof in multicloud realms. This holistic approach to security means that companies can navigate the cloud with confidence, protecting their digital assets for years to come.

Toward Proactive Cloud Security with CNAPP Solutions

Securing applications and data is a real challenge in cloud-native development because its ability to scale as quickly as it does means that the security tests required must be done while accommodating for this rapid application scaling. When APIs form an essential part of communication and data exchange, CNAPP emerges as the go-to solution to deal with AI/ML-driven API complexities. In this section, we cover all the main features and advantages of CNAPP as well as its place in forming future cloud-native security. CNAPP offers a fresh approach to securing cloud-native applications and APIs. It tackles common security myths head-on, promotes a comprehensive security strategy, and delivers forward-thinking solutions. This helps companies stay one step ahead of potential threats across multiple cloud platforms. With CNAPP, businesses can navigate the complex world of cloud computing with greater confidence, knowing their digital assets are well-protected for the long haul.

- **Graph-Based Technology:** CNAPP uses graph-based algorithms to visualize potential attack paths, allowing organizations to understand and prioritize risks effectively. This context-aware analysis helps teams focus on vulnerabilities that pose the most significant threats.
- **Critical Attack Path Identification:** CNAPP consolidates isolated security findings into a comprehensive view of potential attacker movements, enabling organizations to craft more effective security strategies.

- **Prioritization and Remediation:** By understanding the interconnected nature of cloud environments, CNAPP provides prioritized security recommendations and out-of-the-box remediation solutions, including dynamic deny guardrails.

- **Proactive Security Posture:** CNAPP empowers security teams to secure cloud infrastructures proactively, reducing the need for reactive responses and improving overall security readiness.

- **Continuous Discovery and Inventory:** CNAPP automatically discovers APIs and builds a current inventory of them so that you know what to protect. Visibility is crucial for security teams to evaluate what APIs might have gone undocumented or been forgotten so that all can be watched.

- **Identification of Vulnerabilities:** CNAPP identifies vulnerabilities in APIs throughout all preproduction and production environments via cutting-edge scanning technologies. With early detection, teams can prioritize the vulnerabilities that are more likely to be exploited and thus reduce their risk of exposure.

- **Automated Policy Enforcement:** CNAPP empowers organizations to define and automatically enforce security policies on all APIs. That ensures a consistent state of security posture as APIs evolve and comply with both internal standards and regulatory compliance.

- **Deep Inspection and Anomaly Detection:** CNAPP can detect abnormal activities using real-time API traffic, such as data exfiltration or unauthorized access. With this level of scrutiny at the depths, organizations are better equipped to rapidly respond and lessen overall damage from attacks.

- **Integrated Security in CI/CD Pipeline:** CNAPP integrates security insights into development pipelines, fostering a shift-left approach that embeds compliance and security from day one.

- **Advanced API Risk Management:** CNAPP enables granular view of the security state and hints at optimal prioritization. This enables companies to focus on the most significant risks, better protect their overall resiliency, and reduce attack surface exponentially.

Enhancing API Security with LLMs and CNAPP Solutions

The rapid evolution of cloud-native ecosystems demands intelligent tools that can monitor and protect vast, dynamic environments. Large language models (LLMs), such as GPT, have emerged as critical allies in this domain. Their ability to process and analyze massive data sets aligns perfectly with the needs of API security in modern cloud-native architectures.

The integration of LLMs aligns seamlessly with CNAPP's zero trust model, ensuring that no interaction is trusted by default. Every API call undergoes dynamic scrutiny, allowing intelligent, context-aware decisions to mitigate threats in real time.

- **Real-Time Anomaly Detection:** LLMs embedded within CNAPP solutions analyze API traffic patterns to detect subtle irregularities, such as abnormal spikes, unusual sequences, or unauthorized data movements.

- **Sophisticated API Abuse Mitigation:** Attack vectors like API scraping or response manipulation can be identified through the deep pattern recognition capabilities of LLMs.

- **Proactive Policy Enforcement:** By continuously learning from traffic patterns, LLMs help refine API security policies over time, ensuring compliance with zero trust principles.

CNAPP solutions implement the zero trust security model by continuously verifying API access and enforcing strict security policies. This approach ensures that only authorized users interact with sensitive resources, effectively safeguarding cloud-native applications from emerging threats.

- **CVE Focus vs. Holistic Security:** While addressing common vulnerabilities and exposures (CVEs) is important, CNAPP advocates for a more comprehensive "inside-out" approach. Rather than focusing solely on known vulnerabilities, CNAPP emphasizes securing the broader cloud environment to defend against both known and emerging threats.

- **Cloud Workload Protection Platform (CWPP):** Although CWPPs are critical, CNAPP highlights the need for proactive solutions that address security before breaches occur, advocating for a multifaceted defense strategy.
- **Least Privilege in the Cloud:** Traditional least-privilege models face challenges in cloud environments. CNAPP suggests that even minimal permissions can lead to significant damage in cloud-based attacks, calling for more adaptive access control practices tailored to the cloud.
- **Private vs. Public Cloud Security:** CNAPP dispels the myth that private networks within public clouds are inherently secure. The platform emphasizes that all cloud assets, regardless of their network designation, are accessible through cloud provider APIs, making robust security essential across the board.

Summary

In this chapter, we covered the foundational strategies of securing cloud-native applications—a transition that involves moving away from old, perimeter-based strategies and adopting a more integrated, proactive approach toward application security. We also talked about DevSecOps and shift-left practices, which means bringing security as early as possible in the development to minimize vulnerabilities before hitting production.

We covered the compliance and cybersecurity tools provided by the Cloud Native Computing Foundation (CNCF), to help ensure containerized applications and microservices are secure during runtime while also supporting the agility of cloud-native environments.

We then turned toward securing applications in serverless architectures, where boundaries are not really present at an infrastructure level, and introduced function-level permissions and API gateway protection, and the need to ensure that dependencies are periodically updated as a means to prevent software from being the attack vector that allows vulnerabilities to get into production.

In an effort to both empower users with the ability to claim and control digital identities, and bridge the traditional world with the emerging high-risk world of Web3, we explored the applicability of Web3 technologies—particularly decentralized identity (DID)—as a means to transfer the control of identity away from companies, and also to bring that parallel within the greater context of a zero trust model. We also explored large language models (LLMs), like GPT, which, when brought together with tools like CNAPP, provide an additional layer of API security via actionable real-time anomaly identification and automated policy enforcement.

References

1. "OWASP Cloud-Native Application Security Top 10": https://owasp.org/www-project-cloud-native-application-security-top-10/
2. "OWASP Top Ten 2025": https://owasp.org/www-project-top-ten/
3. "The Ten Most Critical Risks for Serverless Applications v1.0": https://github.com/puresec/sas-top-10
4. CNCF Landscape: https://landscape.cncf.io/?group=projects-and-productsandview-mode=grid
5. "Using RBAC Authorization": https://kubernetes.io/docs/reference/access-authn-authz/rbac/
6. "CNCF Official Guide": https://landscape.cncf.io/guide#introduction
7. "OWASP API Security Project": https://owasp.org/www-project-api-security/
8. "OWASP Serverless Top 10": https://owasp.org/www-project-serverless-top-10/
9. Serverless-Goat: https://github.com/OWASP/Serverless-Goat

Chapter 17. Data Center Segmentation On-Prem to the Cloud

In this chapter, you will learn about the following:

- The role of segmentation in hybrid and multicloud environments, including leveraging granular access control and continuous validation for secure zero trust and microsegmentation
- Challenges of hybrid and multicloud segmentation, including addressing inconsistent policies, scaling issues, and visibility gaps in dynamic environments
- End-to-end segmentation with unified policies to ensure consistent policy enforcement across on-premises and cloud platforms
- Segmentation migration from on-premises to cloud automation and orchestration for policy management, centralizing visibility and threat detection across segmented networks
- Cloud-native, open-source, and centralized segmentation solutions for hybrid environments
- The impact of Web3 on segmentation and its relation to zero trust

Introduction to Data Center Segmentation in Hybrid and Multicloud Environments

The era of relying on a single data center or cloud provider to deliver better solutions for modern IT has passed. Applications have moved away from stationary locations in this era of hybrid and multicloud computing. Instead, they run in a distributed fashion that uses a combination of on-premises infrastructure, public and private cloud, and edge computing resources. While this transformation offers tremendous flexibility and scalability, it also introduces major challenges related to security and operations.

In the face of these challenges, *segmentation*, which is the practice of logically dividing IT environments to enforce security boundaries, has become one of the cornerstones of security and compliance strategies. Segmentation minimizes attack surfaces, prevents lateral movement, and keeps sensitive data safe by isolating workloads and limiting access on granularity-based policies. Yet, conventional segmentation based on VLANs, firewalls, and access control lists (ACLs) simply will not scale as organizations evolve to hybrid and cloud-native architectures. This is mainly due to the operational complexity and overhead of managing segregated segmentation systems and multiple policy models, each limited to its own specific domain.

The Limitations of Traditional Segmentation

Classic solutions work well against systems that have a static, on-premises structure, where the network border can be clearly defined and is pretty stable. VLANs aggregate resources, firewalls limit traffic, and ACLs govern access. Or in the case of the cloud, different cloud providers implement their own proprietary segmentation models (such as AWS Security Group). Overall, while providing basic enforcement capabilities within their own domains, these tools are unable to cope with the cloud-native realities of multicloud businesses:

- **Ephemeral Workloads:** Containers or serverless functions come and go in seconds, making static configurations impractical.
- **Distributed Architectures:** Applications cross many domains with them mandating policies that can extend seamlessly across on-premises, cloud, and edge environments.
- **Dynamic Traffic Patterns:** East-west traffic—workload communication within a data center or cloud—is far more prevalent, requiring new approaches to monitor and control.

Without more adaptive and scalable segmentation frameworks, organizations face risks such as policy gaps, misconfigurations, and regulatory noncompliance, especially as workloads migrate between environments.

A New Approach to Segmentation

Segmentation needs to adapt to tackle the limitations discussed in the preceding section. Today, modern solutions are based on approaches such as zero trust and microsegmentation that provide fine-grained access controls and continuously verify every interaction in a network. In addition to the good old network constructs, segmentation today encompasses identity-based policies, application-aware controls, and dynamic orchestration that align with the cloud-native architectures.

Key elements of this modern segmentation approach include

- **Unified Policies Across Environments:** These policies are created with a single definition and are applied consistently on-premises, on different cloud domains, and at the edge.
- **Scalability and Automation:** Cloud-native environments demand high scale and speed, typically achievable through orchestration and AI-driven tools.
- **Integration with Security Frameworks:** Segmentation aligned with frameworks like zero trust provides continuous verification and least-privilege access.
- **Emerging Technologies:** New tools—like Web3-based immutable trust models—are used to increase transparency and security in complex architectures by leveraging blockchain and decentralized identity principles. In the context of segmentation, blockchain ensures tamper-proof policy enforcement and audit trails, preserving the integrity of segmentation rules across distributed systems. This approach aligns with zero trust principles by ensuring that every access request is verified and validated through a decentralized trust network, reducing attack surfaces and preventing unauthorized lateral movement across segmented environments.

This chapter explores how organizations can adopt these elements to meet the demands of hybrid and multicloud segmentation. By examining the evolution of segmentation practices, we'll highlight the tools and strategies that provide consistency, scalability, and security in today's distributed IT environments. From foundational concepts like zero trust to advanced capabilities like Cisco Cloud Network Controller (CCNC) and Web3, this chapter aims to equip you with the insights needed to transform segmentation into a strategic enabler of cloud-native security.

Zero Trust and Microsegmentation Principles for Segmentation

Organizations worldwide are grappling with the need to protect ever-expanding critical assets and essential user bases as cyber threats grow larger in scale and complexity. Attackers use known exploitable vulnerabilities to move laterally, and trusted insiders also misuse their access. Static boundaries are disappearing, and the traditional perimeter-based model of security is no longer sufficient, especially in hybrid and multicloud environments. Such a security landscape has led to new security models, including zero trust and microsegmentation, working together in harmony to build resilient defenses against the modern threat landscape.

What Is Zero Trust?

In a nutshell, *zero trust* is based on the mantra "never trust, always verify," an expression coined by Forrester alum John Kindervag in 2009. Zero trust removes the assumption of implicit trust from the network. Regardless of where the user, device, or application is and how many times it has been granted access, now each and every interaction should be authenticated and authorized. Zero trust moves the focus from perimeter security to a model that assumes any network is already compromised while constantly performing real-time validation on access requests.

For a more detailed explanation of the core principles of zero trust, see Chapter 1, "Zero Trust Demystified."

Key Benefits of Zero Trust

Adopting a zero trust framework provides significant advantages, enhancing security visibility, reducing risks, and enabling robust protection across hybrid and cloud-native architectures:

- **Enhanced Visibility:** Delivers granular insights into all activities across workloads, users, and devices.
- **Reduced Lateral Movement:** Prevents attackers from navigating within the network by requiring reauthentication at every segment boundary.
- **Protection Against Insider and External Threats:** Reduces the risk of privilege misuse and data exfiltration.
- **Cloud and Hybrid Support:** Provides robust security for assigned environments, distributed environments, and cloud-native environments that have departed from traditional perimeters.
- **Simplified Security Posture:** Establishes a comprehensive structure that minimizes dependence on separate point products.

The Synergy of Zero Trust and Microsegmentation

As a foundational technology underpinning zero trust, microsegmentation creates connected workspaces in which policy-based, identity-aware workload isolation, services and applications, and dynamic perimeters allow only specifically permitted communication. In contrast to traditional segmentation based on wider network zones, microsegmentation contains more specific access controls and the principle of least privilege, reducing attack surface and lateral movement opportunities for attackers. It improves east-west traffic visibility by ensuring that breaches in compromised segments do not affect the rest of the network while making compliance and audits easier.

This method scales harmoniously within hybrid, cloud-native, and on-premises surroundings, dynamically addressing security requirements without operational disruption. For example, in a distributed healthcare organization, microsegmentation guarantees that the medical records system communicates only with approved billing systems, regardless of whether their workloads live in different clouds. Likewise, hybrid setup users working with Kubernetes clusters are given fine-grained access to exact namespaces with actions, enabling no lateral movement by accident.

Microsegmentation, which was also discussed in Chapter 4, "Security and Segmentation," and in shorthand form throughout this book, operationalizes zero trust by defining workload-specific microperimeters, allowing no lateral movement across the network, and applying a consistent security posture across even the most diverse architectures. Together, these strategies promote multilevel integrated security policies that mitigate risk and adapt to evolving threats over time, consistent with a zero trust approach to perpetual verification and least privilege.

Zero trust microsegmentation is not a technical shift but a strategic imperative in the modern threat landscape. Combined, these methodologies offer a powerful basis for alignment of segmentation principles with contemporary security frameworks with fine-grained control and ongoing validation to ensure security posture is inherently adaptable as cyber attack vectors evolve.

Identity-Aware Segmentation

Today's segmentation is based on workload or user identity policies, not static network characteristics, such as IP addresses. So, this identity-aware approach is right in line with zero trust by making access decisions more contextual and dynamic.

East-West Traffic Control

In hybrid environments, microsegmentation is critical as east-west traffic increases. Organizations can also block lateral movement—a common tactic, technique, or procedure (TTP) used by advanced persistent threats (APTs) a common advanced persistent threat (APT) tactic, technique, or procedure (TTP)—by enforcing communication boundaries between data centers and across clouds.

Dynamic Isolation

The ephemeral nature of cloud-native workloads demands segmentation policies that can adapt in real time. Microsegmentation ensures that whenever workload scaling or migration takes place, the security control can travel along with the workload so that there is consistent enforcement.

Behavioral Monitoring and Adaptive Policies

Such real-time behavioral analytics have been implemented into zero trust architectures to identify deviations. Where adaptive policies are beneficial —for example, isolating a compromised workload or automatically revoking suspicious access—microsegmentation complements this behavior.

Segmentation Challenges in Hybrid and Multicloud Environments

Moving to hybrid and multicloud has provided companies with unparalleled flexibility and scalability by enabling a variety of workloads to run on different combinations of on-premises and cloud architectures. But this adaptability also creates major complications, especially with segmentation. Unified security across these diverse environments means overcoming fragmented policies, visibility gaps, and frequent changes in workload behavior.

Inconsistent Policy Frameworks

Each cloud provider offers its own segmentation tools, such as AWS Security Groups, Azure Network Security Groups (NSGs), and GCP VPC Firewall Rules. These tools work great for their own ecosystems, but they are siloed and require custom synchronization to enforce consistent policies between platforms.

- **Policy Fragmentation:** Use of different terminologies, capabilities, and policies by different providers leads to operational overload and increased chances of misconfigurations.

- **Impact on Zero Trust:** While zero trust principles call for consolidated, identity-oriented policies, fragmented frameworks erode the enforcement of those policies and lend themselves to policy drift, whereby configurations diverge over time.

Consider an enterprise that has a multicloud application, with sensitive data in AWS and compute workloads in Azure. Maintaining consistent segmentation rules across these environments is painfully manual and error-prone, which creates the potential for gaps and weaknesses.

Visibility and Compliance

Achieving 360-degree visibility of network traffic, user interaction, and workload behavior in hybrid environments will forever be a challenge. Due to the absence of centralized oversight, tracing interactions across domains is difficult for organizations, which creates two major challenges:

- **Limited Threat Detection:** Threat actors can leverage gaps in domain crossover to move laterally or exfiltrate data.
- **Compliance Complexities:** Regulations, such as General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA), require sensitive data to be protected at all costs. Each instance of data loss triggers an audit process that is time-consuming and inefficiently complex due to fragmented visibility.

Being able to see everything from one place requires integrating disparate tools and ensuring that monitoring spans both east-west traffic (within clouds) and north-south traffic (across environments).

Dynamic Workloads

Containers and serverless functions are examples of cloud-native workloads that are ephemeral by nature. Static segmentation models simply will not handle the issues because their lifecycle can be measured in seconds or minutes.

- **Scalability Challenges:** Policies must scale automatically and at runtime as workloads are created, terminated, or moved between environments.
- **Policy Application Lag:** Workloads change constantly, and manually updating segmentation policies leads to delays in policy application, which translates into loss of protection.

For example, a containerized microservice deployed to Kubernetes may spin up dozens of instances during peak traffic. If segmentation policies are not automatically applied to these instances, they remain vulnerable.

While such scenarios create new opportunities, they also pose an equally large number of questions, or rather challenges. Cloud-native techniques using software-defined networking (SDN) and virtual private clouds (VPCs) help by abstracting configurations from the underlying network. However, they still require integration into a broader, unified segmentation strategy to truly support multicloud and hybrid environments.

Ways to Address These Challenges

Modern segmentation models solve these problems by abstracting policies from the underlying infrastructure, allowing for common enforcement across environments. Tools such as Cisco Cloud Network Controller (CCNC) provide an excellent example of this approach through the generation of an application-centric policy model that covers hybrid and multicloud environments.

- **Consistency Across Platforms:** Cisco Cloud Network Controller allows organizations to define policies once and enforce them consistently across on-premises, private, and public cloud environments. This capability ensures security posture uniformity, regardless of infrastructure complexity.
- **Visibility and Monitoring:** Integrated monitoring provides centralized visibility, breaking down operational silos and allowing security teams to detect and respond to threats across domains efficiently.

- **Dynamic Policy Enforcement:** Automated response mechanisms enable policies to adapt to the dynamic nature of cloud-native workloads, maintaining security without manual intervention.

Integrating segmentation within a wider zero trust framework enables organizations to avoid these problems and ultimately achieve security and compliance in complex hybrid environments.

Ways to Implement End-to-End Segmentation Policies with Zero Trust

With the growing trend of hybrid and multicloud architectures, how do organizations secure their workloads consistently, at scale, and without millions of lines of code to rewrite? Principles of zero trust act as a framework for guiding responses, championing ongoing verification, least privilege, and identity-based access controls. Solutions such as Cisco Cloud Network Controller allow organizations to operationalize end-to-end segmentation policies based on these principles.

Unified Policy Definition Across Domains

In a zero trust architecture, every workload has its own identity and access control, so each range of things is treated as a unique object. Policies are dynamically enforced according to workload characteristics such as behavior, risk profile, and compliance requirements.

For example, a policy might specify that a billing system may talk to an inventory system only if certain conditions are met. Since these systems run in Azure, AWS, or on-premises, the same policy applies, and we no longer have gaps in enforcement.

Workload-Specific Segmentation

In a zero trust architecture, every workload has its own identity and access control, so each range of things is treated as a unique object. Policies are dynamically enforced according to workload characteristics such as behavior, risk profile, and compliance requirements.

Examples of workload-specific segmentation include

- **Production vs. Development:** Workloads in production may require tighter segmentation policies than workloads for development, which are more permissive.
- **High-Risk Workloads:** For sensitive applications that process personal data, they could be isolated from less critical systems and be monitored and restrained.

By tailoring policies to individual workloads, organizations can minimize attack surfaces and enhance regulatory compliance.

Methods to Prevent Policy Drift

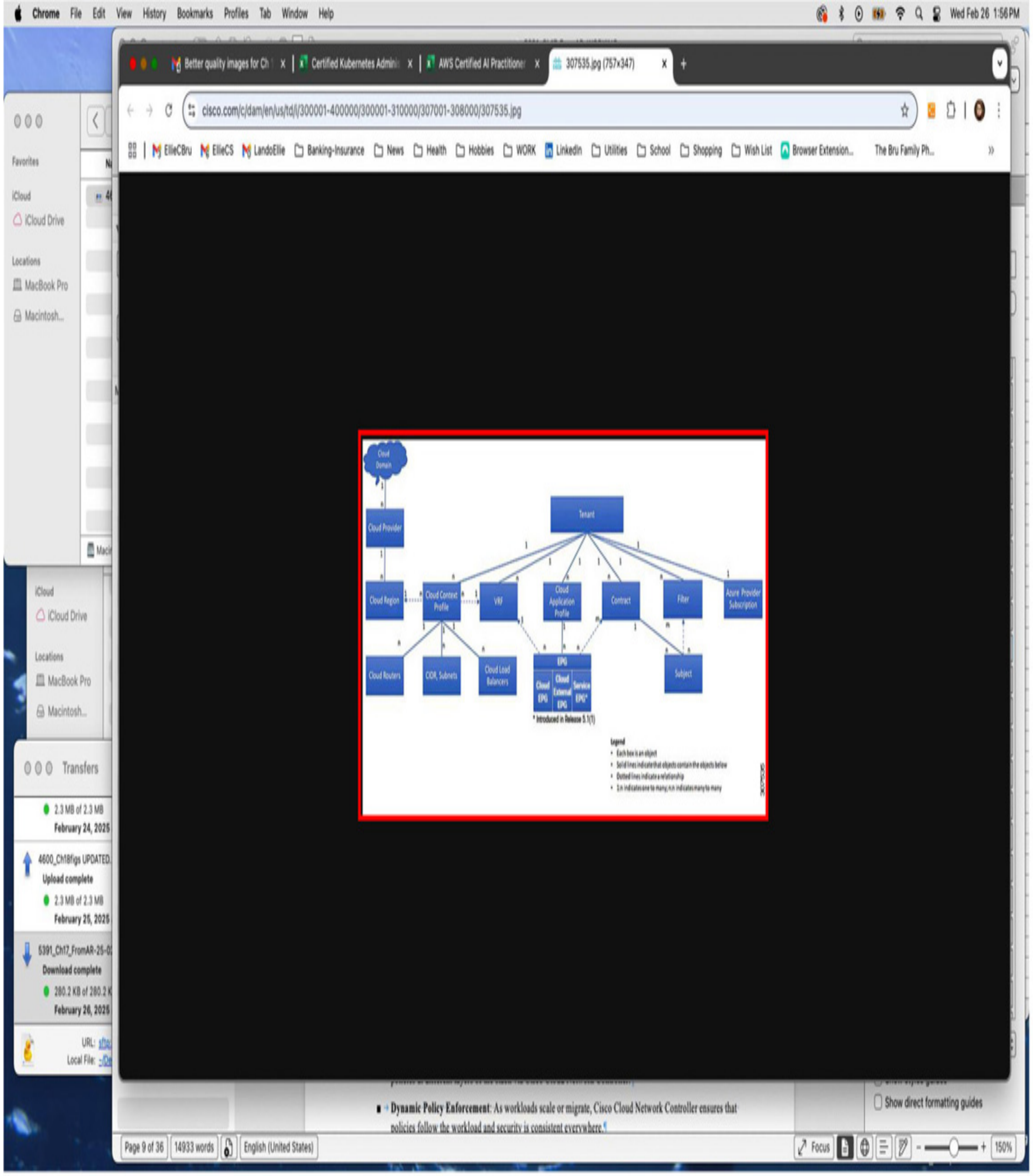
Inconsistent enforcement or manual updates across environments can leave configuration policies diverging, a phenomenon known as *policy drift*. And, in hybrid environments where workloads shift between domains, this represents a major risk.

- **Role of Automation:** Tools like Cisco Cloud Network Controller synchronize policies automatically, ensuring consistent enforcement across all environments.
- **Proactive Monitoring:** With telemetry in place, any deviation in the application of a policy can be detected in real time, allowing administrators to take preventive action before problems set in.

If, for example, a rule that blocks access to a sensitive database is incorrectly configured in one cloud, automated synchronization makes sure that change is corrected immediately.

Core Features of Cisco Cloud Network Controller for Unified Segmentation

At the heart of Cisco Cloud Network Controller is an object-oriented policy model that abstracts policy from the underlying infrastructure (see Figure 17-1). It also allows administrators to design dynamic, application-aware policies that easily extend across hybrid and multicloud environments:



* Introduced in Release 5.1(1)

Legend

- Each box is an object
- Solid lines indicate that objects contain the objects below
- Dotted lines indicate a relationship
- 1:n indicates one to many; n:n indicates many to many

- **Dynamic Policy Enforcement**: As workloads scale or migrate, Cisco Cloud Network Controller ensures that policies follow the workload and security is consistent everywhere.

Figure 17-1 *Cisco Cloud Network Controller Object Model and Relationships*

- **Application-Centric Segmentation:** Policies attach to applications, not static network constructs, so they are portable across environments. One example is a policy regarding access between a CRM and a database; such a policy is relevant regardless of whether these components are hosted on-premises or in the cloud.
- **Service Insertion:** Integration with third-party tools such as firewalls, intrusion detection systems (IDS), and load balancers enforces policies at different layers of the stack via Cisco Cloud Network Controller.
- **Dynamic Policy Enforcement:** As workloads scale or migrate, Cisco Cloud Network Controller ensures that policies follow the workload and security is consistent everywhere.

The Role of Zero Trust in Unified Policies

Zero trust strengthens unified segmentation strategies by delivering consistent security enforcement across hybrid and multicloud environments. A zero trust model continuously validates the access request in real time and allows only little access needed to maintain least-privilege access. Such fine-grained control limits lateral movement in segmented environments and contains breaches. In a convergence segmentation strategy, it indicates consistent, adaptive security policies that seamlessly span any underlying infrastructure. Should a workload be compromised, zero trust policies can immediately isolate it, without affecting other segments and thus preserving the integrity of the environment as a whole.

The Cisco Cloud Network Controller enables this capability by abstracting the policies into one single model that is independent of platforms. The same single unified API target is easier to manage because policy can be defined once and universally applied to on-premises and cloud. Centralized policy enforcement complements zero trust by ensuring visibility and control over segmented traffic flows, and dynamically altering policy based on whether workloads scale, move, or change behavior. It also makes it easier to reach or enforce zero trust from a segmentation perspective because some of the complexity of classification and enforcement have now been offloaded across clouds, into a homogeneous operating cloud model.

Ways to Migrate Segmentation Policies: From On-Premises to Cloud

Migrating segmentation policies from on-premises to the cloud is far more than a simple replication exercise. Cloud architectures differ fundamentally, with ephemeral workloads, dynamic communication patterns, and provider-specific tools. To ensure security continuity and compliance, segmentation policies must evolve to align with cloud-native paradigms and adhere to zero trust principles.

Beyond policy replication, organizations must reimagine segmentation to accommodate distributed architectures and dynamic workloads. Achieving multisite consistency and ensuring end-to-end policy enforcement requires centralized tools, robust frameworks, and seamless integration across domains.

Adapting Segmentation to Cloud-Native Architectures for Zero Trust Integration

Historically, segmentation for on-premises workloads was enabled using static constructs based on VLANs, IP addresses, and ACLs. In contrast, cloud-native environments require identity-based policies that are dynamic and consistent across domains. These policies enable least-privilege access by tying access controls to workloads and users rather than static IPs or network boundaries. They are dynamic, responding to changes in workload location, scale, or behavior in real time, and maintain a unified security posture across hybrid and multicloud environments.

For example, an on-prem database behind VLANs may require access segmentation using AWS Security Groups, Azure NSGs, or Kubernetes Network Policies post-migration. Cisco Cloud Network Controller abstracts these policies and translates them into cloud-native formats, ensuring consistency across environments. This abstraction simplifies management and enhances security by consistently enforcing policies over hybrid and multicloud environments.

Successful segmentation migration replaces static configurations with identity-aware policies that recognize workload identity and role, allowing for fine-grained access controls. This aligns with zero trust principles by implementing least-privilege access according to workload or user identity and maintaining a “never trust, always verify” approach.

Continuous verification detects anomalies in real time, mitigating threats at the earliest opportunity. This maintains zero trust’s security posture in dynamic, hybrid, and multicloud environments. By implementing Cisco Identity Services Engine (ISE), organizations can dynamically apply security group tags (SGTs) based on workload roles, user characteristics, or risk scores, ensuring hop-by-hop segmentation policy enforcement.

These identity-based policies automatically adjust to workload-agnostic changes in location, size, or behavior. For example, when workloads scale out to new cloud regions, policies automatically secure the new instances without manual intervention. This ensures protection wherever workloads go—a fundamental characteristic of zero trust.

Navigating the Transition from Cisco Cloud Controller

Although Cisco Cloud Controller has reached end-of-life status, its conceptual architecture remains valuable for organizations still relying on it or looking to understand core principles of policy-driven cloud migration. Throughout this chapter, references to Cisco Cloud Controller serve as a mental model for application-centric segmentation rather than a recommendation for any specific product. The primary takeaway is the declarative approach to managing segmentation and security policies, which can be applied to a variety of modern solutions, including open-source platforms and newer Cisco offerings.

Cisco’s decision to discontinue Cloud Controller represents a shift toward unifying zero trust and multicloud segmentation under a broader, more integrated suite. Rather than maintaining a single-purpose solution, Cisco now channels its efforts into offerings like

- **Cisco Identity Services Engine (ISE)** for centralized policy enforcement.
- **Cisco Nexus Dashboard Orchestrator (NDO)** for cross-cloud orchestration.
- **Cisco VXLAN GPO Cloud Segmentation** for fine-grained microsegmentation in VXLAN-based fabrics.
- **Cisco Multicloud Defense** and **Cisco SD-WAN OnRamp** to further extend security and segmentation capabilities to public clouds and branch edges.

Because product lines and technology roadmaps can change over time, you should always consult your Cisco representatives for the most current information on available solutions.

Meanwhile, the principles of segmentation that underpin Cisco ACI—most notably its application-centric policies and declarative model—remain just as relevant. ACI's unified policy framework helps maintain consistent security across on-premises and public cloud infrastructures. This continuity of architectural philosophy ensures organizations can adopt a holistic security posture while migrating to or operating in multicloud environments.

Overview of Policy Models: Cisco ACI and Cloud-Native Constructs

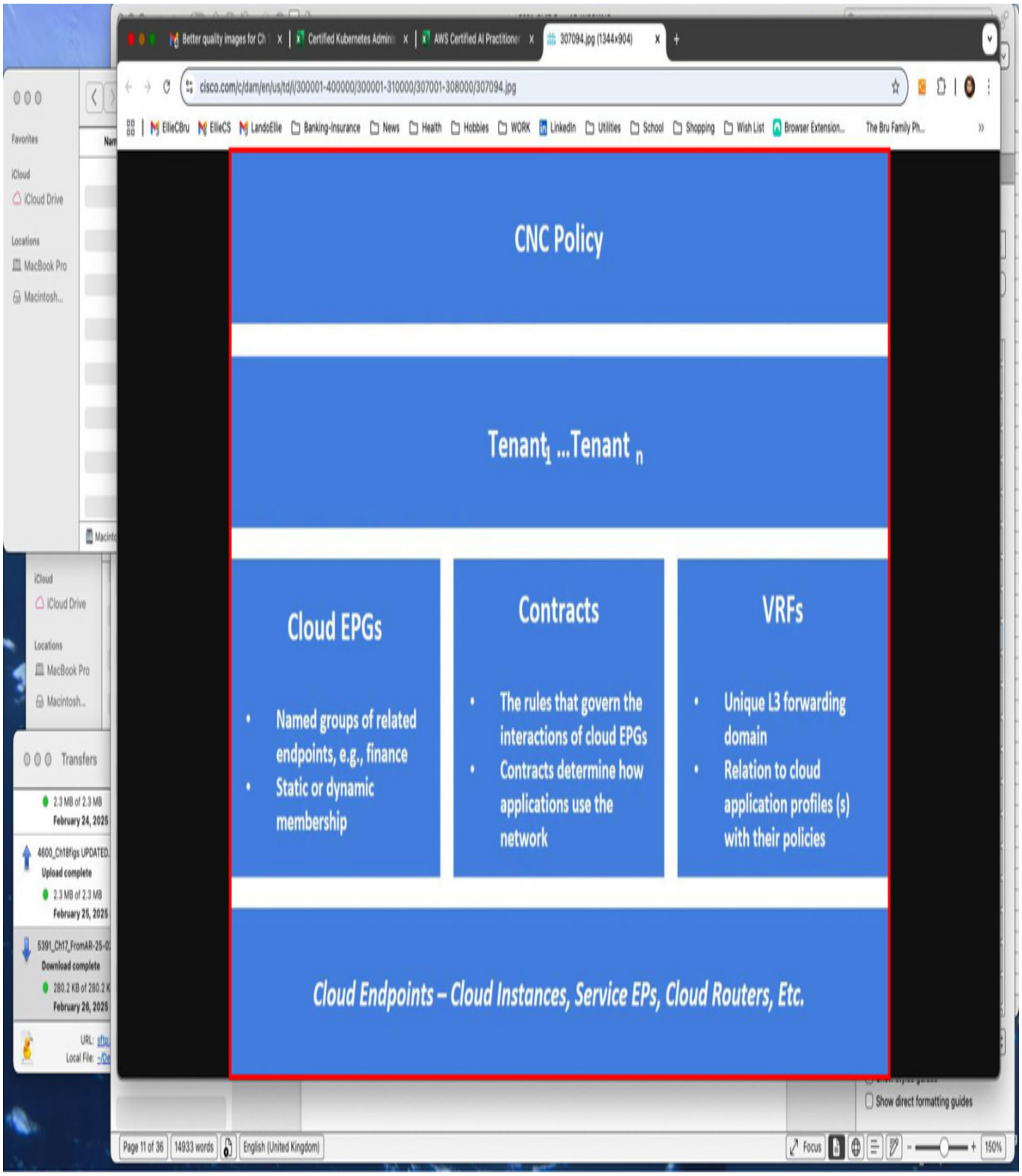
Even with Cisco Cloud Controller discontinued, the broader concepts of unified policy management, logical grouping, and declarative definitions remain at the heart of segmentation solutions. Cisco ACI and major cloud providers (AWS, Azure, GCP) all use group-based policy models to control resource communication. Although each environment has its own constructs, the core ideas—logical segmentation, role-based controls, and centralized policy management—are consistent.

Cisco ACI Policy Model

Cisco ACI organizes network policies and configurations into logical constructs that enforce consistent network policy and segmentation across environments:

- **Tenants:** These are logical containers for isolating policies and resources with RBAC enforcement. A tenant can represent an organization, department, or logical grouping.
- **Bridge Domains (BDs) and Subnets:** These Layer 2 forwarding constructs have defined IP spaces, providing intra-tenant isolation.
- **Endpoint Groups (EPGs):** These are logical groupings of endpoints (such as VMs or containers) based on function or role. Policies between EPGs are defined using *contracts*.
- **Contracts and Filters:** These define and enforce communication rules between EPGs by specifying permitted traffic types.
- **VRFs (Virtual Routing and Forwarding):** The routing domains provide Layer 3 segmentation and isolation, often mapped to cloud-native constructs like VPCs or VNets.

Figure 17-2 illustrates the unified policy model of Cisco Cloud Network Controller, showing how network policies are abstracted into tenants, bridge domains, EPGs, contracts, and VRFs so that security is consistently enforced across on-premises and cloud deployments.



Better quality images for Ch
Certified Kubernetes Admin
AWS Certified AI Practitioner
307094.jpg (1344×904)
cisco.com/c/dam/en/us/td/i/300001-400000/300001-310000/307001-308000/307094.jpg
Favorites
iCloud
iCloud Drive
Locations
MacBook Pro
Macintosh...
Transfers
Page 11 of 36
14933 words
English (United Kingdom)
Show direct formatting guides
Focus
150%

CNC Policy

Tenant$_1$ ...Tenant $_n$

Cloud EPGs

- Named groups of related endpoints, e.g., finance
- Static or dynamic membership

Contracts

- The rules that govern the interactions of cloud EPGs
- Contracts determine how applications use the network

VRFs

- Unique L3 forwarding domain
- Relation to cloud application profiles (s) with their policies

Cloud Endpoints – Cloud Instances, Service EPs, Cloud Routers, Etc.

Figure 17-2 *Cisco Cloud Network Controller Unified Policy Model*

This abstraction allows administrators to define security policies at a higher level of intent rather than configuring each endpoint individually. Figure 17-3 illustrates how the Cisco Cloud Network Controller implements an application-centric policy model that abstracts endpoints into logical groups known as endpoint groups (EPGs). In this model, contracts are used to specify security definitions, governing communication between EPGs by explicitly defining the rules, protocols, and traffic flows permitted between them.

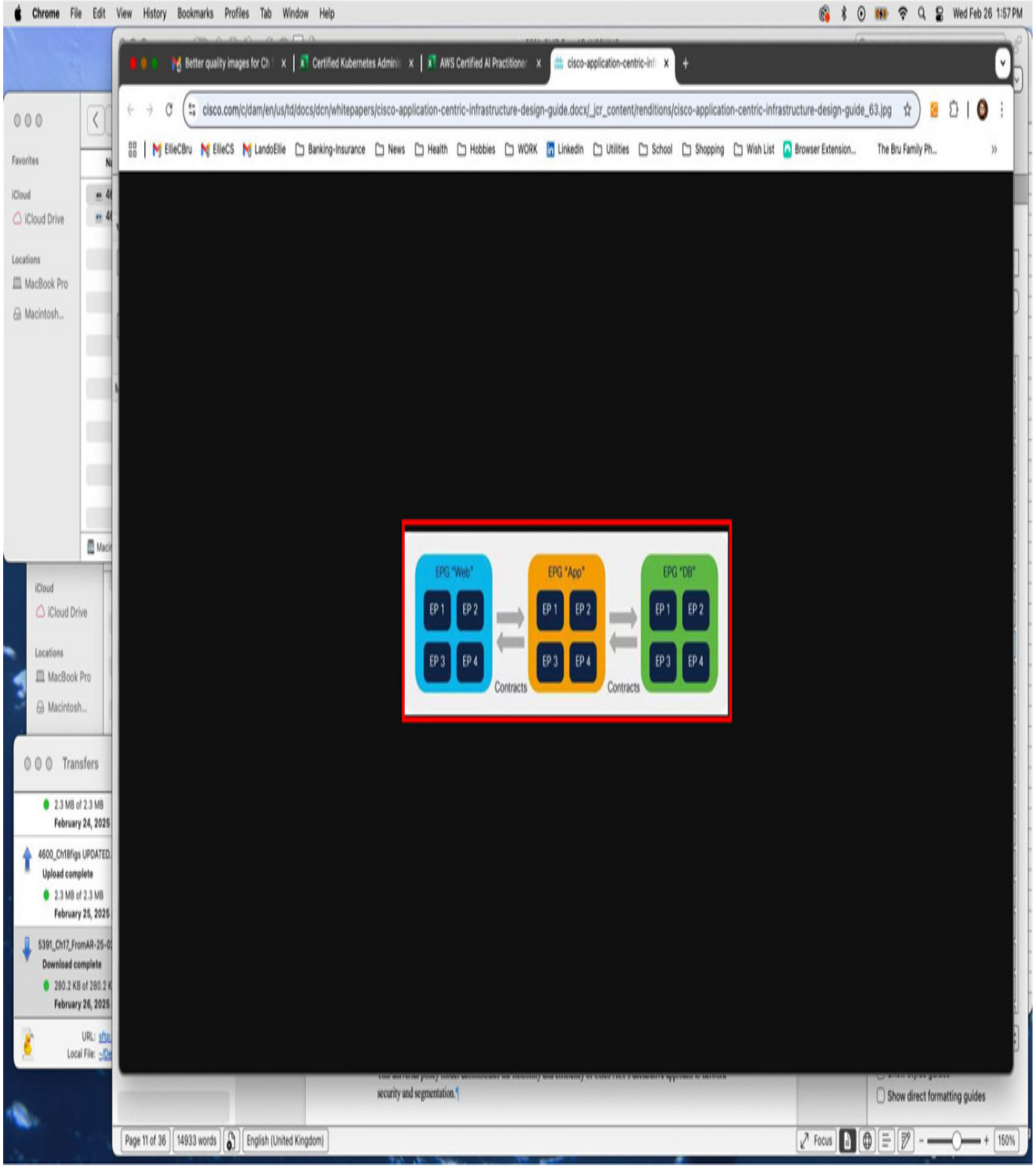


Figure 17-3 *Cisco Cloud Network Controller Universal Policy Model*

This architecture decouples policy definitions from the underlying infrastructure, enabling consistent enforcement across on-premises and cloud environments. By centralizing security and segmentation rules within contracts, the model simplifies automation, enhances scalability, and maintains a robust security posture aligned with zero trust principles. This universal policy model demonstrates the flexibility and efficiency of Cisco ACI's declarative approach to network security and segmentation.

Cloud Policy Models (AWS, Azure, GCP)

Major cloud providers implement comparable constructs for managing and securing communication between resources:

- **Logical Containers (Accounts, Projects, Subscriptions):** Define boundaries for organizing and isolating resources with RBAC.
- **Network Segmentation (VPCs, VNets):** Enable Layer 3 constructs defining IP address spaces and routing domains for resources.
- **Security Grouping (ASGs, Tags, Labels):** Enable logical groupings of resources for applying access and security policies.
- **Traffic Filtering (Firewall Rules, Security Groups):** Define inbound and outbound communication rules, similar to ACI's contracts.

Figure 17-4 provides an overview of how Cisco Cloud Network Controller translates its policy model into these cloud-specific configurations, creating a consistent, application-centric approach across multiple platforms. It highlights the translation of logical containers, network segmentation constructs, and security groupings into cloud-native terms.

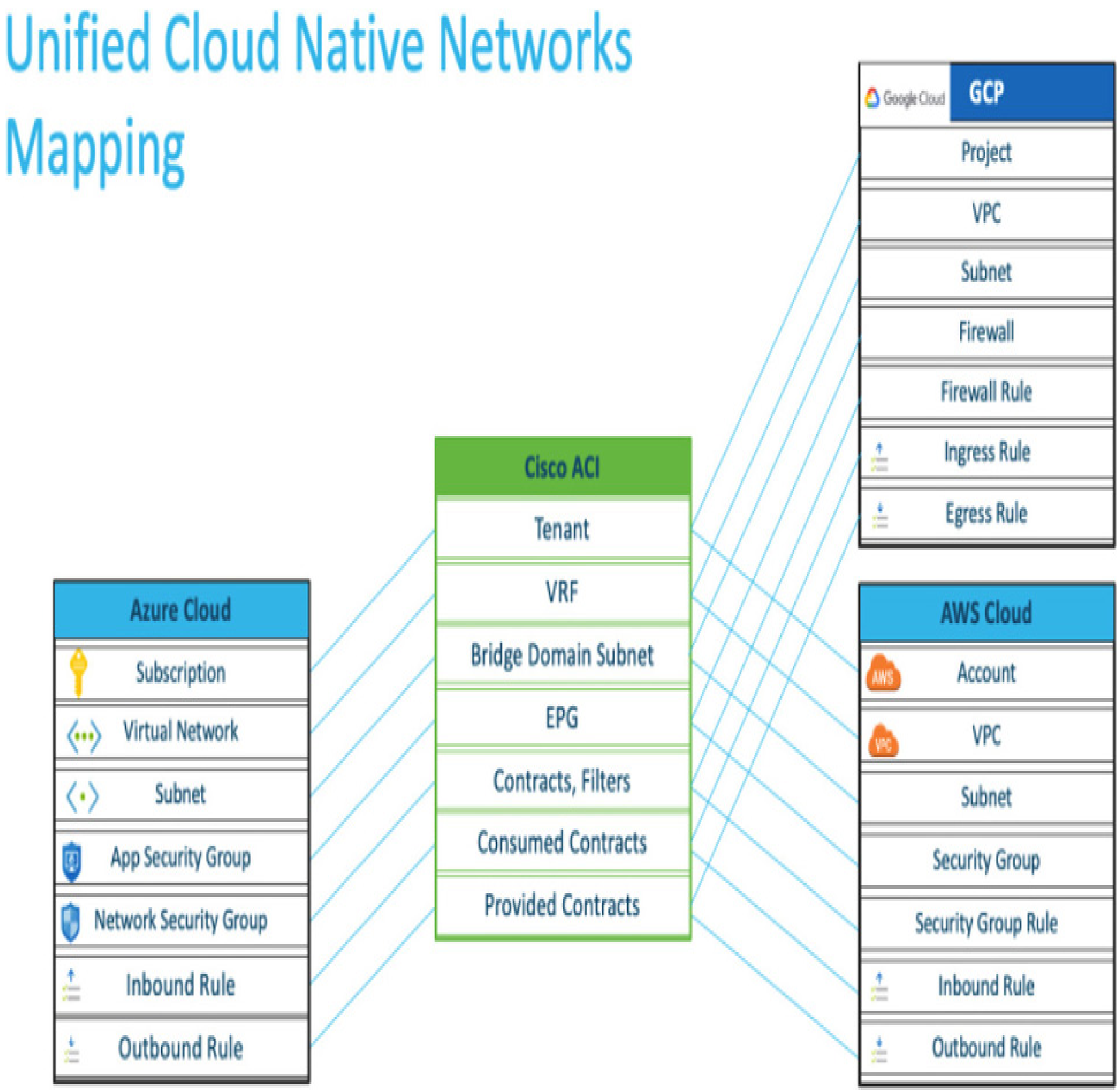


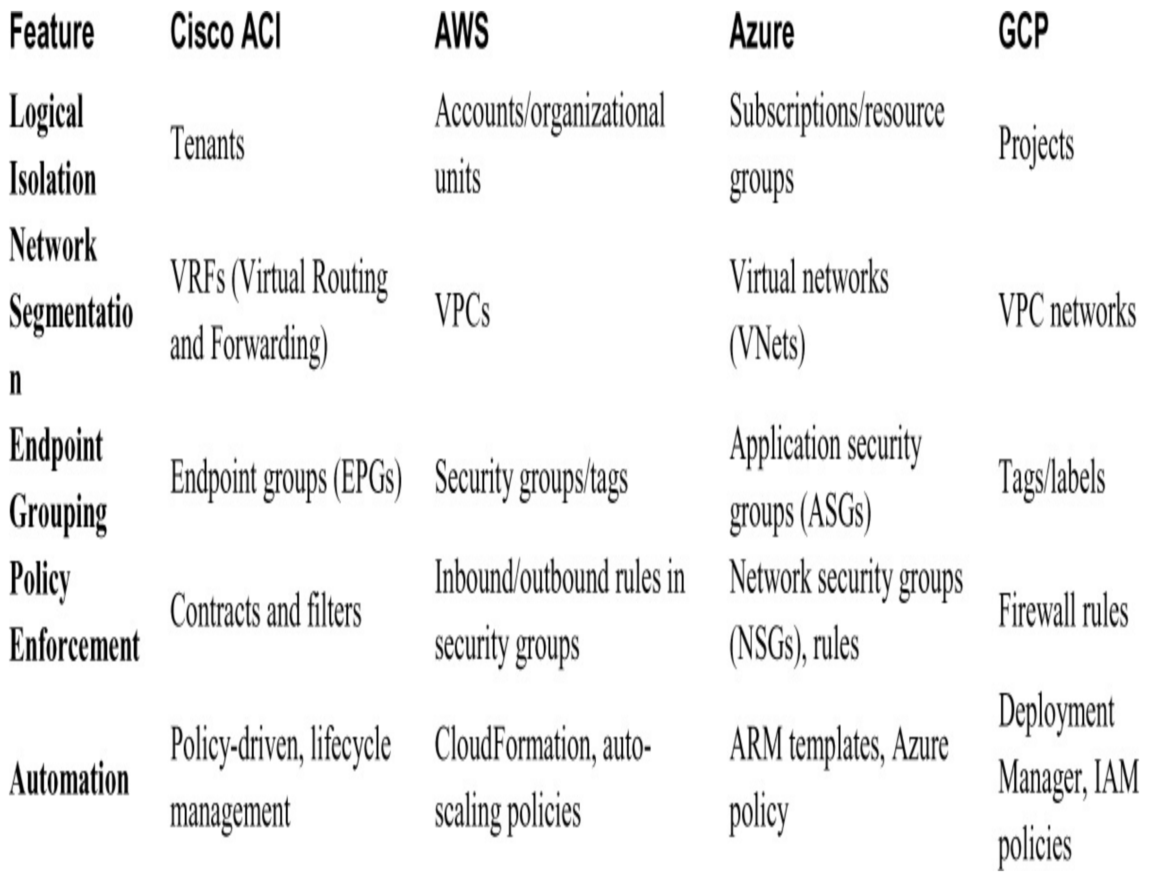
Figure 17-4 *Cisco Cloud Network Controller Unified API Mapping*

Note: As the Cisco Cloud Network Controller was developed as a sub-product that adheres to the ACI policy approach but is designed for cloud environments, we refer to ACI as a holistic policy framework. Depending on the context, we use Cisco ACI and Cisco Cloud Network Controller interchangeably to reflect their alignment within the same policy model.

Comparison and Mapping of Policy Constructs

Table 17-1 compares key policy constructs in Cisco ACI with those of AWS, Azure, and Google Cloud. Despite variations in naming conventions and implementation details, the underlying principles of segmentation, access control, and policy-driven management remain consistent.

Table 17-1 *Comparison of Cisco ACI and Cloud Policy Constructs*



Feature	Cisco ACI	AWS	Azure	GCP
Logical Isolation	Tenants	Accounts/organizational units	Subscriptions/resource groups	Projects
Network Segmentation	VRFs (Virtual Routing and Forwarding)	VPCs	Virtual networks (VNets)	VPC networks
Endpoint Grouping	Endpoint groups (EPGs)	Security groups/tags	Application security groups (ASGs)	Tags/labels
Policy Enforcement	Contracts and filters	Inbound/outbound rules in security groups	Network security groups (NSGs), rules	Firewall rules
Automation	Policy-driven, lifecycle management	CloudFormation, auto-scaling policies	ARM templates, Azure policy	Deployment Manager, IAM policies

Comparison to Cloud Segmentation Models

Cisco ACI’s contract model aligns closely with cloud-native segmentation but offers unique advantages in policy management and automation. In many cloud platforms, segmentation uses security groups, firewall rules, and network security groups, each requiring explicit inbound and outbound rules. In contrast, Cisco ACI uses *contracts* that define both provider and consumer roles, automatically generating bidirectional ACL filters. This reduces the complexity of managing traffic directionality and ensures consistent enforcement across hybrid and multicloud environments.

In cloud models, role assignments are typically based on source (initiator) and destination (receiver). By contrast, Cisco ACI explicitly defines consumer and provider roles in contracts. This separation enhances policy reuse and reduces administrative overhead. While cloud-native tools often require separate rules for ingress and egress, Cisco ACI contracts unify both directions under one configuration, simplifying policy management and ensuring consistency.

Contracts as Security Policy Objects in Cisco ACI

In Cisco ACI, contracts are foundational for defining communication rules between endpoint groups. They establish provider and consumer roles and dictate ACL filtering direction, aligning with cloud-native segmentation practices. Contracts manage traffic based on source, destination, ports, and protocol-based filtering, enabling intuitive and abstract policy models with granular control.

Provider-Consumer Model in Cisco ACI

The contracts communication model enforces and automatically creates the security policies with awareness of direction during the filter implementation. This dual-filter model enforces stateful policy rules without requiring separate ACLs. A basic example would look like this:

- **Consumer:** Entity initiating communication (such as a Web EPG making HTTP requests).
- **Provider:** Entity receiving communication (such as an App EPG responding).

A single contract between a Web EPG (consumer) and an App EPG (provider) specifies allowed ports and protocols, governing ACL filtering in both directions:

- **Consumer-to-Provider:** Filters traffic based on the destination port, allowing only authorized incoming requests.
- **Provider-to-Consumer:** Filters return traffic based on the source port, maintaining stateful communication.

Contracts are implemented using subconstructs called *subjects* and *filters*:

- **Subjects and Filters:** A subject is a construct contained within a contract and references a filter. A contract contains one or more subjects, and a subject contains one or more filters. A filter contains one or more filter entries. A filter entry is a rule specifying fields such as the TCP port and protocol type. Figure 17-5 provides a logical visualization of a single filter construct, which has two entries for HTTP and HTTPS.

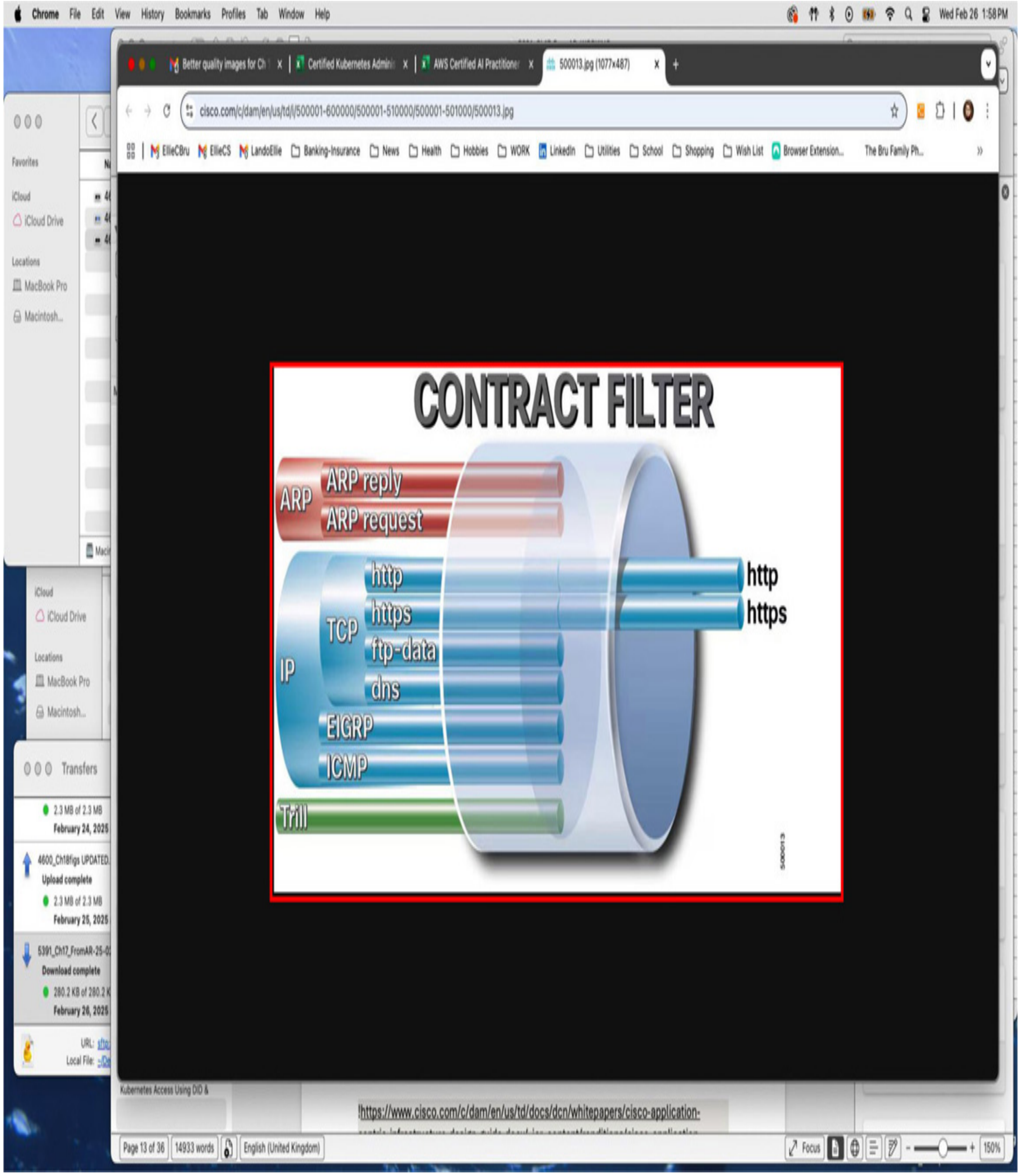


Figure 17-5 *Contract Filter*

Figure 17-6 illustrates the relationship between contracts, subjects, and ACL filters in Cisco ACI’s architecture (in this figure, only one filter entry is expressed).

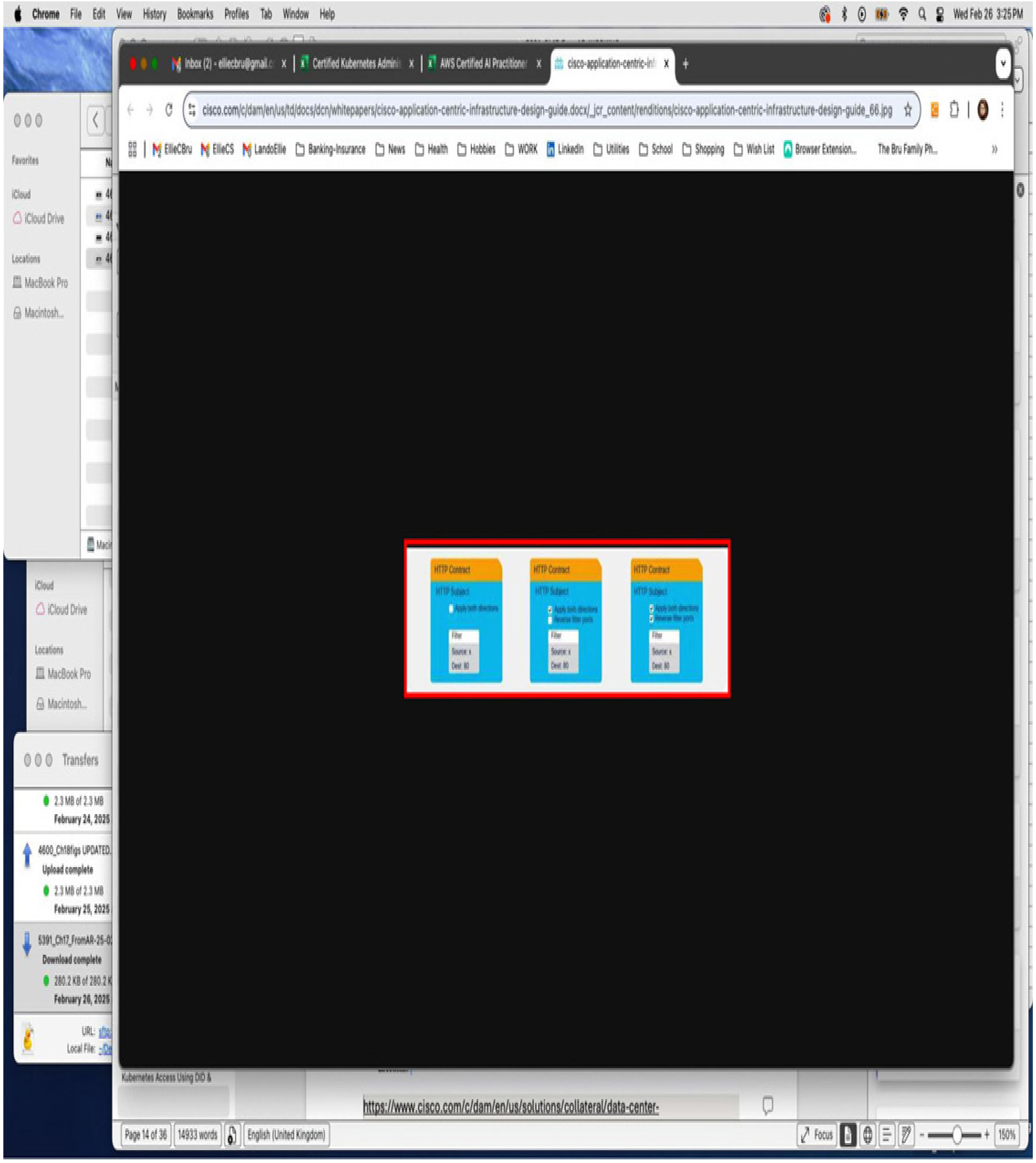
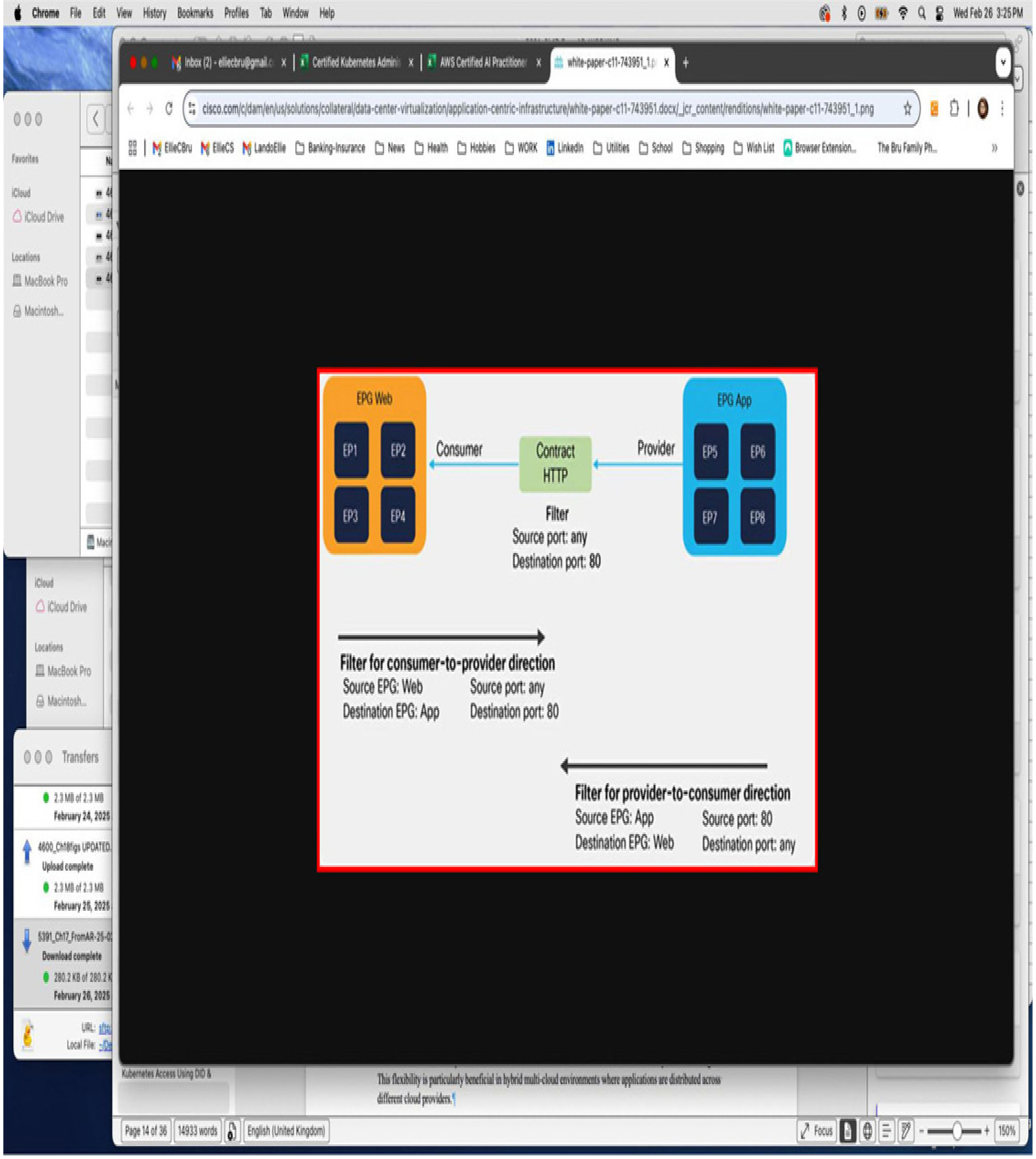


Figure 17-6 *Cisco Cloud Network Controller Contracts Subjects and ACL Filters*

Creating a contract object provides a granular and flexible mental model for policy definition. Now all that’s left is to associate the contract among workloads. By applying the contract policy between EPGs, you can essentially terminate the process of policy implementation today and in the future when additional endpoints may be instantiated. This is due to the fact that the policy is implemented by the identity context of the workloads in the cloud. Figure 17-7 demonstrates how a single contract with an HTTP filter (source port: Any, destination port: 80) automatically deploys two filters for both directions:



This flexibility is particularly beneficial in hybrid multi-cloud environments where applications are distributed across different cloud providers.

Figure 17-7 *Web as Consumer and App as Provider*

How Cisco ACI Contracts Simplify Network Management

Cisco ACI contracts unify ingress and egress traffic rules into a single, bidirectional configuration. This approach reduces configuration complexity and minimizes the risk of misaligned security rules. With protocol- and port-specific filtering, Cisco ACI provides precise access management across EPGs. Reusable contracts promote consistency and remove duplication, streamlining policy creation and maintenance across on-prem, cloud, and hybrid environments. By abstracting complex ACL configurations into logical contracts, Cisco ACI supports scalable microsegmentation and zero trust security, aligning seamlessly with modern cloud-native best practices.

One of the key advantages of Cisco ACI is its ability to facilitate communication within an endpoint group without requiring contracts, even when endpoints reside in different cloud environments. For instance, an AWS instance in the Web EPG can seamlessly communicate with an Azure VM in the same Web EPG without any contract configuration. This flexibility is particularly beneficial in hybrid multicloud environments where applications are distributed across different cloud providers.

When an EPG is stretched across multiple environments, Cisco Nexus Dashboard Orchestrator (NDO) automatically deploys the necessary routing and security policy configurations in each environment. This automation reduces the operational burden on administrators and ensures consistent security policies across cloud platforms.

Despite the differences in routing implementations between AWS and Microsoft Azure, Cisco’s Multi-Cloud Networking solution normalizes this complexity by providing a unified network and security policy model. This consistency simplifies operations and enhances scalability in hybrid multicloud architectures. Administrators can create virtual routing and forwarding instances (VRFs), EPGs, and contracts between EPGs on the NDO. Through integration with Cisco Application Policy Infrastructure Controller (APIC) and Cloud Network Controller, these configurations are automatically deployed across all environments, ensuring consistent policy enforcement and reducing the risk of misconfiguration.

The ability to stretch applications across multiple cloud sites while maintaining consistent network and security policies is a crucial feature of Cisco ACI that any multicloud operation should aspire to. By abstracting multicloud networking into logical constructs, organizations are able to deploy applications across hybrid environments without compromising on security or operational efficiency and without the overhead of coordinating or maintaining siloed models.

Figure 17-8 illustrates the deployment of an application stretched across multiple cloud sites within a single tenant environment.

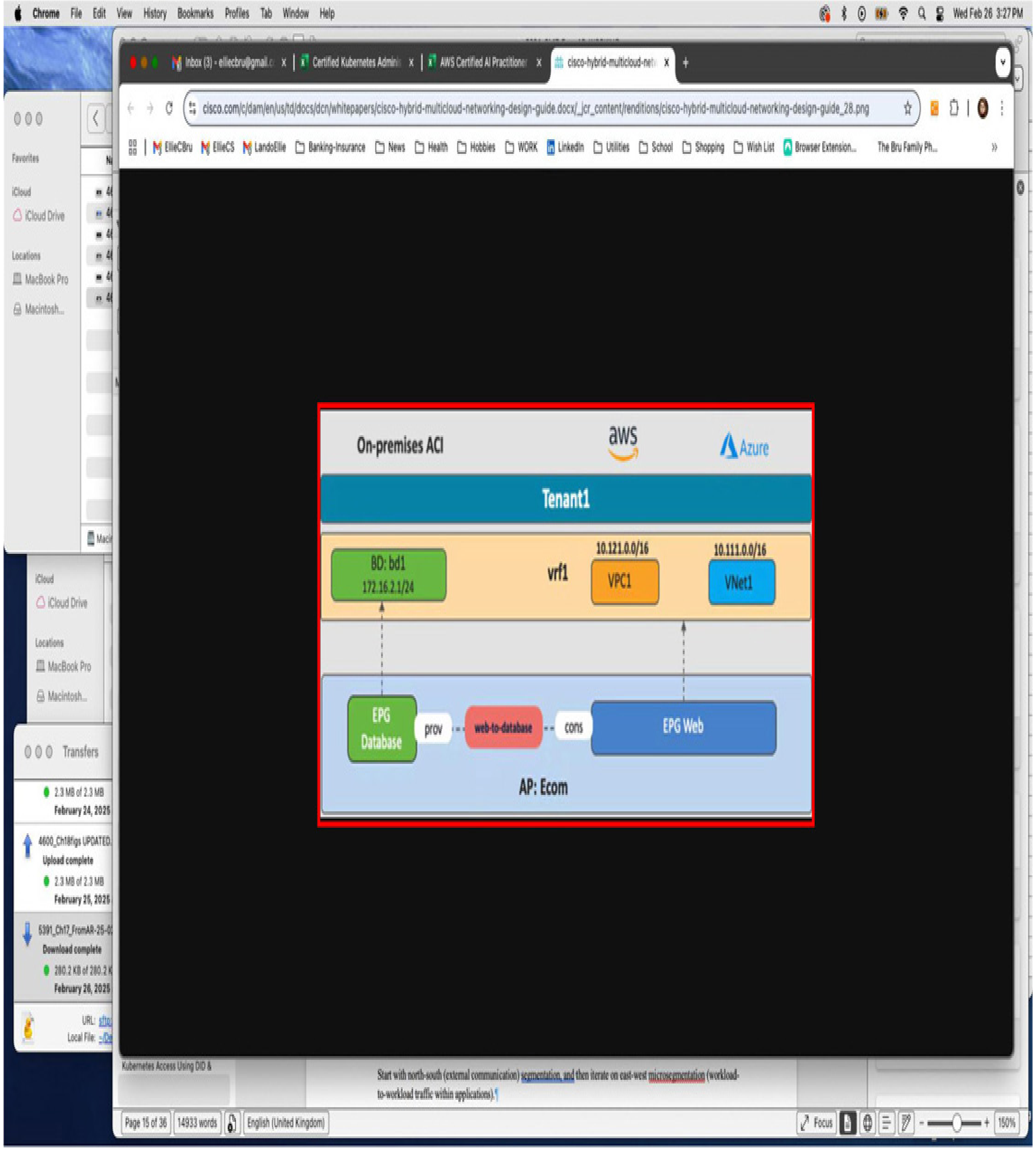


Figure 17-8 *Application Stretched Across Sites (Intra-Tenant)*

The simplified resulting state showcases how Cisco ACI's unified policy model facilitates seamless communication between endpoints in different environments while maintaining consistent security policies.

Effective Segmentation Migration Methods

Migrating segmentation policies requires adopting strategies that align with dynamic, cloud-native environments and zero trust principles. A successful migration translates static, network-centric constructs into dynamic, application-aware policies in accordance with zero trust principles. Balancing security with the shift to cloud-native architectures takes a transition strategy.

Centralized management tools are important in reducing risk during segmentation migration. These tools enable policies to be synchronized in real time, effectively between on-premises and cloud environments to enforce consistency as new environments transition. Also, conducting pre-cutover testing to ensure segmentation policies are validated is also standard.

This approach allows companies to validate compliance and operational integrity of security rules before the full migration by simulating traffic flows. Doing so is important because migration can cause problems that may lead to downtime, but with this proactive testing phase, you can determine exactly where issues are and prepare for a better trajectory while migrating and keeping security posture responsive. Organizations that want to maintain security and operational consistency while providing a smooth transition should approach segmentation migration systematically. Migration patterns are described in the following sections.

Phased Migration with Policy Layers

An organization can start with north-south (external communication) segmentation and then iterate on east-west microsegmentation (workload-to-workload traffic within applications).

For instance, an identity-based policy that limits external access to a communication either ingress or egress can be a starting point to allow only specific external entities to access workloads. Once this foundational layer is established, the next step is to strengthen security by implementing granular isolation at the application tier. This can be achieved using security groups or Kubernetes network policies to enforce stricter workload-to-workload communication controls.

Parallel Deployment for Hybrid Continuity

An organization can enforce segmentation policies at the same time in both on-premises and cloud environments with parallel deployments. This method prevents having a gap in enforcement while the system migrates. Tools such as Cisco Cloud Network Controller can propagate segmentation rules between on-premises data centers and cloud domains. That allows the organization to transition seamlessly with policies being applied in both environments.

Automated Policies for Rapid “Big Bang” Migrations

In this approach, organizations can opt for a full cutover with pre-validation and automated policy enforcement so that workloads are secured immediately after they are live in the cloud. Automation tools can reduce the complexity of this approach by configuring and enforcing the policies on a real-time basis, thereby mitigating human errors, which tend to cause delays and misconfigurations.

Consistency Across Hybrid and Multicloud Environments

Maintaining consistent segmentation policies across multiple clouds and on-prem sites is a major challenge. Solutions like Cisco Cloud Network Controller and Cisco Nexus Dashboard Orchestrator offer a centralized control plane to define, synchronize, and enforce segmentation rules. Similar capabilities can also be achieved with open-source orchestration tools or other vendor-agnostic platforms that provide centralized management for creating and updating policies, automation for real-time synchronization of workload changes, and multisite or tenant-based frameworks that logically segment regions and clouds without manual rule duplication. These tools help to ensure seamless security, operational agility, and compliance across heterogeneous environments by maintaining consistent segmentation policies

Figure 17-9 illustrates a brief topology example of Cisco Multi-Cloud Networking connecting an on-premises ACI fabric and two cloud sites.

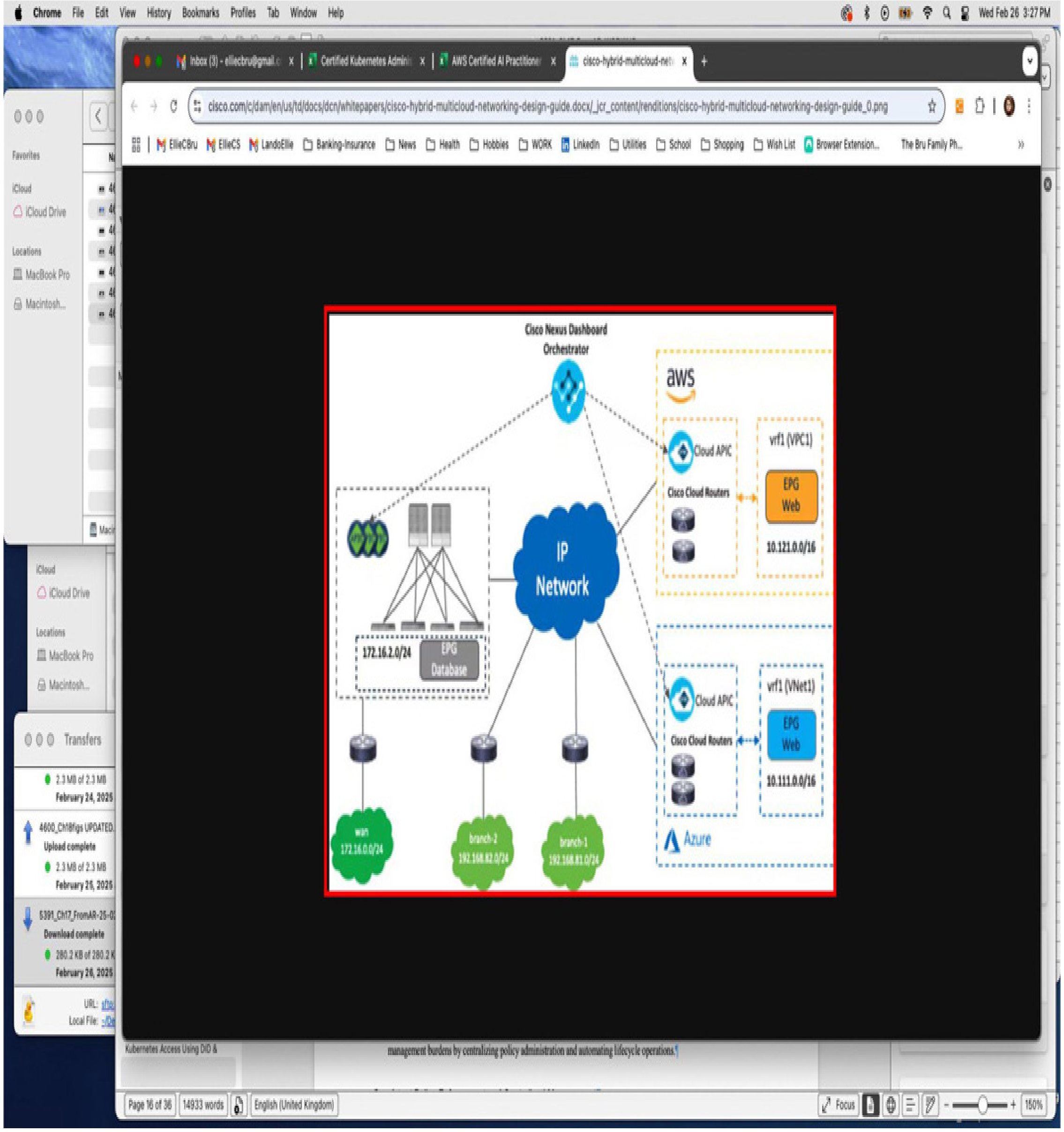


Figure 17-9 *Cisco Multi-Cloud Networking Connecting an On-Premises ACI Fabric, an AWS Environment, and a Microsoft Azure Environment*

How to Leverage Tenants for Multisite Management

Cisco ACI’s tenant-based architecture simplifies multisite management by providing policy isolation, centralized control, and robust role-based access control (RBAC). Tenants are logical containers that encapsulate policy definitions, enabling organizations to manage policies across departments, teams, and regions while maintaining governance and security consistency.

This approach extends across multiple data centers or cloud environments, ensuring seamless segmentation and zero trust boundaries.

Tenants enhance policy management through granular RBAC, which restricts who can modify policies within each tenant, and policy isolation, preventing accidental crossover of rules between different business units. Additionally, automated scaling ensures that tenant-based segmentation adjusts dynamically as workloads expand, maintaining zero trust boundaries without manual intervention.

By centralizing control and enabling policy isolation, tenants simplify the complexities of multisite management, ensuring consistent policy enforcement across hybrid and multicloud environments. Organizations can define and enforce segmentation policies within a single tenant or across multiple tenants representing specific business units or regions, ensuring uniform segmentation policies across clouds and on-premises environments while maintaining isolation between development, testing, and production environments.

This dynamic policy enforcement and adaptation allow policies to automatically adjust to workload changes, location shifts, or scaling requirements. By leveraging tools like Cisco Cloud Network Controller and Cisco Nexus Dashboard Orchestrator, organizations achieve operational efficiency and automation, reducing manual management burdens by centralizing policy administration and automating lifecycle operations.

Consistent Policy Enforcement and Centralized Management

Tenants abstract segmentation policies, ensuring consistent application across diverse environments without the need for manual replication. By leveraging tenant-based RBAC, organizations can enforce governance at a granular level, restricting resource access to relevant teams or regions. This approach ensures uniform policy enforcement by consistently applying segmentation rules across clouds and on-premises environments while maintaining isolation between development, production, and testing environments.

Enhanced governance is achieved by ensuring that specific teams or regions can only access authorized resources, strengthening overall organizational security. In a multisite environment, tenants are integral in keeping policies consistent and isolated. This allows all security policies to be unified, in a synergistic and cohesive way, between on-premises and cloud environments. It also ensures that teams in one region do not accidentally gain access to resources in another region, maintaining security boundaries and compliance in distributed environments.

Dynamic Policy Enforcement and Adaptation

Tenants in Cisco ACI enable dynamic policy enforcement across multisite architectures. Policies automatically adapt to changes in workload behavior, location, or scale. This dynamic nature ensures that segmentation rules are consistently applied across cloud providers, while allowing different environments to be segmented from one another. It also supports global governance, ensuring compliance by maintaining security boundaries across distributed architectures.

For example, as workloads scale to a new region, policies automatically secure the new instances without manual intervention. This dynamic adjustment capability is essential for maintaining a consistent security posture in hybrid and multicloud environments. It reinforces zero trust principles by continuously verifying and protecting workloads regardless of their location.

Operational Efficiency and Automation

Tenant-based management streamlines segmentation policy deployment through automation tools like Cisco Cloud Network Controller and Cisco Nexus Dashboard Orchestrator. These solutions centralize policy management and automate lifecycle operations across distributed environments, improving efficiency while reducing misconfigurations. The automation framework enables seamless scaling and policy adaptation across multiple regions and clouds, maintaining security compliance as organizations grow. Whether using Cisco proprietary or open-source tools, organizations should aim for a model where they define policies once and enforce them programmatically, thus preventing configuration drift and human error as they expand their cloud presence.

Multisite Application of Tenants

For multisite architectures, tenants provide unified segmentation management, adding significant value to complex, multifaceted organizations. Tenant policies can span multiple interconnected sites, ensuring consistent segmentation across a larger architecture. By enabling automation, tenants remove operational overhead while maintaining consistent policy enforcement across clouds, simplifying the scaling process.

Additionally, tenants allow for site-agnostic policy extension, where segmentation rules are enforced across geographically dispersed environments. This approach reduces friction when deploying new tenants while maintaining governance and ensuring that security boundaries are preserved. It also enables centralized control planes to manage agile cloud environments effectively.

How to Use a Multicloud Policy Orchestrator Across Clouds

Migrating to a multicloud environment while maintaining consistent segmentation policies is a complex task. Cisco's Nexus Dashboard Orchestrator streamlines this process with a single-pane-of-glass view to manage and orchestrate policy between sites, including entire hybrid on-premises data centers and cloud stacks. NDO facilitates the integration of new cloud sites into the infrastructure as well as the application of the same segmentation policies across all sites.

Building secure and reliable network connectivity is the first step that organizations should take to integrate a new cloud site to their on-premises data center. Before onboarding the new site into NDO, organizations should establish the necessary infrastructure components (virtual networks, subnets, security groups, and so on). After connecting to NDO, the organization can add the new site by entering the necessary information such as credentials and API endpoints. NDO and cloud site communication is verified in the dashboard; then integration is completed.

Figure 17-10 illustrates the Cisco ACI Multisite Architecture utilizing Cisco NDO to synchronize policies using a centrally managed versioned-controlled configuration schema.

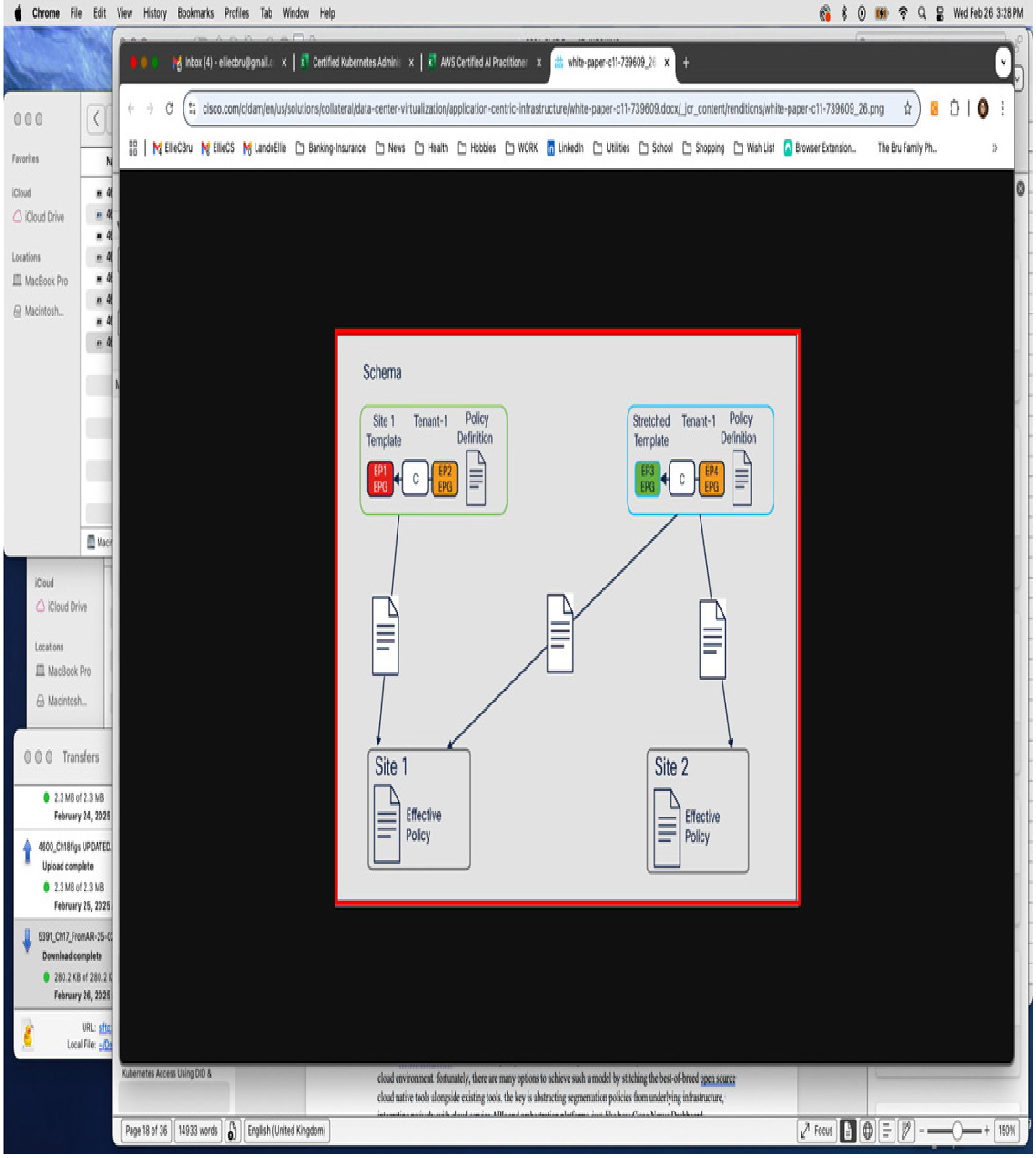


Figure 17-10 *Cisco ACI Multisite Architecture for Consistent Policy Enforcement*

NDO allows for the import of existing templates from on-premises environments to ensure consistent segmentation policies. Such templates must be reviewed and adjusted so that they complement the configuration and specific needs in the new cloud environment. Templates are then applied to the new cloud site in NDO's Schemas section, and policies are pushed and monitored for compliance.

After deployment, consistent segmentation across sites is validated through rigorous testing and traffic monitoring. NDO’s analytics provide insights into traffic patterns, confirming that segmentation effectively isolates workloads and restricts unauthorized access. To ensure ongoing security, regular audits of segmentation policies are recommended. As infrastructure evolves, templates should be updated to reflect new requirements, with policies redeployed to maintain consistent enforcement across all environments.

Open-Source Cross-Domain Orchestration and Automation for Segmentation Policies

As hybrid and multicloud architectures grow in complexity, effective management of segmentation policies requires advanced orchestration and automation. Static and siloed security models struggle to keep pace with the dynamic nature of cloud-native environments. Therefore, centralized control planes capable of synchronizing policies across on-premises, public cloud, and edge environments are essential. Let’s explore more advanced strategies for automating and orchestrating consistent segmentation policies across diverse platforms, leveraging a holistic approach that could integrate both Cisco solutions and open-source components.

Following the application-centric policy models of Cisco ACI and cloud-native solutions, advanced automation strategies should definitely provide a more abstract policy model that is not dependent or limited by the constructs of the destined cloud environment. Fortunately, there are many options to achieve such a model by stitching the best-of-breed open-source cloud-native tools alongside existing tools. The key is abstracting segmentation policies from underlying infrastructure, integrating natively with cloud service APIs and orchestration platforms, just like Cisco Nexus Dashboard Orchestrator does.

Most organizations combine multiple tools—cloud-native, open-source, and commercial—to achieve a secure multicloud segmentation strategy but often lack cross-cloud consistency, requiring manual synchronization that increases the risk of misconfigurations.

Integration of Open-Source Tools for Cross-Cloud Policy Consistency

To maintain consistency and reduce operational complexity, organizations must define segmentation policies, deploy them, and maintain them as code. This Infrastructure-as-Code (IaC) approach enables organizations to synchronize policies across hybrid and multicloud environments while maintaining zero trust principles.

- **Ansible** and **Terraform:** These tools provide IaC templates to synchronize segmentation rules across hybrid and multicloud environments. They integrate with Cisco ACI constructs and cloud-native constructs such as AWS Security Groups, Azure NSGs, and GCP Firewall Rules, ensuring consistent policy enforcement across logical containers like tenants, VPCs, and VNets.
- **Crossplane:** This tool extends the automation layer by managing cloud-native resources through Kubernetes-native APIs, allowing organizations to declaratively control segmentation policies alongside application deployments. This approach enhances operational agility and aligns with the Kubernetes-native segmentation models discussed earlier.
- **Pulumi:** This tool adds flexibility by defining infrastructure and policies in general-purpose programming languages such as Python, Go, and JavaScript. This integration enhances CI/CD pipelines and seamlessly connects with GitOps workflows for continuous policy validation.

Open-source tools augment the declarative policy models of Cisco ACI and cloud-native solutions by providing advanced identity-based microsegmentation, service mesh integrations, and API-aware security policies. These solutions ensure consistent segmentation across heterogeneous environments and offer flexibility in highly customized deployments.

- **Calico:** This tool extends Kubernetes network policies with advanced identity-based controls, enabling dynamic, label-based segmentation. It integrates seamlessly with ACI’s endpoint groups and cloud-native security groups, ensuring consistent policy enforcement across hybrid and multicloud deployments.

- **Istio:** This tool enhances application-centric segmentation by abstracting communication between microservices. It provides identity-aware routing, mutual TLS, and policy-based traffic management. Istio enforces zero trust communication across distributed microservices while maintaining consistent security policies defined at the application layer.

- **Consul:** This tool offers service discovery and identity-based segmentation for microservices, aligning with the logical constructs of Cisco ACI and cloud-native VPCs. Its integration with HashiCorp Vault enhances segmentation policies with dynamic secrets management, ensuring that security boundaries adapt to workload identities and behavioral changes.

- **Cilium:** This tool leverages eBPF technology to enforce API-aware microsegmentation, translating high-level policies into platform-specific configurations. It bridges Cisco ACI contracts with cloud-native security constructs like Kubernetes network policies and cloud provider firewall rules.

- **Linkerd:** This tool provides transparent zero trust communication and policy enforcement. This lightweight service mesh focusing on simplicity and minimal resource consumption is particularly suited for edge environments and lightweight microservices architectures.

These open-source tools can complement Cisco’s approach by abstracting infrastructure details beneath a higher-level policy model. They also natively integrate with cloud-native solutions and can help bridge the gap between cloud-native constructs and hybrid environments. Applying them consistently across different types of resources such as Kubernetes clusters, virtual machines, and cloud-native services helps organizations maintain a unified security posture, while harmonizing cloud-native security models with on-premises policies.

AI/ML for Adaptive Policy Enforcement in Hybrid Cloud Environments

In dynamic cloud environments, static policies often lag behind changing workloads or traffic patterns. Cisco Cloud Network Controller integrates artificial intelligence (AI) and machine learning (ML) capabilities to enable adaptive policy enforcement, ensuring security scales with evolving cloud dynamics. By analyzing traffic flows, user behavior, and system interactions, AI/ML models detect anomalies that could indicate threats or misconfigurations.

Modern cloud environments evolve faster than static security policies can track. AI/ML-driven solutions enable continuous adaptation, proactively detecting anomalies and enforcing zero trust principles based on real-time telemetry. While Cisco's products illustrate one implementation, many open-source and commercial AI/ML platforms provide comparable benefits.

For teams seeking an open-source–centric path, various AI/ML frameworks can also deliver policy enforcement that adjusts in real time. These solutions embody the same adaptive security philosophy, using continuous analysis and automated action to maintain a dynamic zero trust posture across on-prem and multicloud domains.

AI/ML-driven solutions—whether part of Cisco's Cloud Network Controller or assembled from open-source projects—are transforming network security by delivering adaptive policy enforcement. By continuously analyzing real-time telemetry, detecting anomalies, and proactively mitigating threats, security teams can maintain a robust zero trust posture across on-premises, hybrid, and multicloud environments.

Open-Source AI/ML Integrations for Adaptive Security

Several open-source and commercial platforms can power AI/ML-driven adaptive policies. While Cisco's products offer one approach, these examples demonstrate how open community-driven tools can be leveraged in a similar way:

- **Kubeflow:** This tool integrates AI/ML pipelines into Kubernetes-based applications, automating anomaly detection and adaptive policy controls.

- **Prometheus + Thanos + Grafana:** These tools collect metrics, store them long term, and visualize behavioral patterns. When combined with anomaly detection, this stack can trigger real-time segmentation rule updates to isolate threats.
- **Open Policy Agent (OPA):** This policy engine ingests AI/ML insights for dynamic decision-making. It can automatically revise or enforce rules based on current threat levels or deviations from normal behavior.

These open-source integrations allow organizations to adopt best-of-breed tooling, enabling them to leverage hosted commercial services or purely open-source solutions. While these tools serve as flexible frameworks that can be customized with various other tools, the core objective remains the same: adaptive, real-time policy enforcement that keeps pace with evolving workloads and threat landscapes.

Anomaly Detection and Real-Time Policy Adjustments

AI/ML models continuously monitor network traffic and workload behaviors, identifying deviations from established baselines. If an application communicates unexpectedly with an unauthorized external service, AI-driven detection flags the anomaly. The system then autonomously tightens policies—such as isolating the workload—until further analysis occurs.

This real-time responsiveness is crucial in modern, fast-evolving cloud environments. Dynamic adjustments to microsegmentation rules and network traffic policies enable organizations to neutralize potential threats before they escalate. By continuously detecting deviations, AI/ML models ensure that security postures remain accurate and agile.

Proactive Threat Mitigation

Beyond anomaly detection, AI/ML proactively mitigates threats by autonomously triggering security measures, such as blocking suspicious traffic, tightening microsegmentation rules, or quarantining compromised workloads. This hands-free approach reduces mean-time-to-contain while preserving a zero trust ethos of ongoing verification.

Investing in integrating AI/ML models into the security and cloud operations is not a trivial task. However, organizations that invest in such a venture could automate better threat responses, allowing security teams to focus on strategic analysis and policy refinement instead of manual interventions for each alert.

Visibility, Monitoring, and Predictive Analytics in Hybrid or Multicloud Environments

Maintaining full-stack visibility across hybrid and multicloud environments is essential for effective adaptive policy enforcement. Whether using commercial solutions or open-source observability stacks, centralized monitoring enables security teams to track segmentation policies, traffic patterns, and compliance status. This comprehensive visibility ensures that policies effectively block unauthorized traffic while adhering to internal and regulatory requirements.

- **Centralized Logging:** Organizations can consolidate logs from on-premises switches, cloud firewalls, service meshes, and container orchestration layers. Real-time correlation across these sources provides a unified view of network activity, enhancing policy effectiveness and incident investigation.
- **Flow Analytics:** AI/ML-driven flow analytics analyze traffic patterns across different segments, identifying unusual behaviors at an early stage. Rapid anomaly detection allows for timely threat isolation, minimizing potential impacts.
- **Continuous Compliance:** Automated checks against regulatory frameworks (such as PCI DSS, HIPAA) ensure that adaptive policies consistently maintain compliance. Any deviation or misconfiguration is instantly flagged for remediation.
- **Predictive Analytics:** By examining historical data, predictive analytics forecast potential threats, identifying emerging patterns and correlating them with known risks. With this intelligence, security

teams can proactively adjust security controls, such as revising microsegmentation rules to limit lateral movement or enhancing access controls based on anticipated threat activities. This knowledge also feeds into threat intelligence databases, bolstering detection and response capabilities across all security layers.

Achieving proactive cyber defense—staying one step ahead of attackers rather than merely reacting to breaches—strengthens a resilient and adaptive security posture.

Web3 and Immutable Trust in Hybrid Cloud Segmentation

To address the complexities of distributed hybrid and multicloud architectures, organizations must reevaluate traditional trust models to enable consistent and reliable segmentation across hybrid and multicloud environments. Enter Web3 technologies—from decentralized identity systems, blockchain, and smart contracts to the ability to redefine the paradigm through which trust is established, audited, and maintained. Such innovations work in perfect harmony with zero trust principles and support segmentation efforts (decentralized tamper-proof systems for policy enforcement and monitoring), increasing transparency, resilience, and control.

What Is Web3?

Web3 is a new paradigm of decentralized infrastructure built on blockchain technology, removing the need for centralized intermediaries and enabling self-sovereign identities, cryptographically verified transactions, and smart contracts. Unlike Web2, where central entities control identity and access, Web3 introduces decentralized identity (DID) and smart contracts to enforce policies autonomously without relying on a single provider. Smart contracts are self-executing programs stored on the blockchain that automatically enforce predefined rules, ensuring secure and transparent execution of agreements without human intervention.

In the following sections, we'll explore how Web3 technologies transform segmentation security, how they integrate with Kubernetes, and what challenges must be addressed for practical implementation.

Web3 Technologies and Zero Trust: A Symbiotic Relationship

Web3 technologies have the potential to change cloud-native security by decentralizing trust and authentication. Web3 is, by design, not implemented over centralized servers like traditional models of the Internet, and thus, there are fewer single points of failure and enforceable policies that cannot be manipulated. It really complements what zero trust stands for and reaffirms:

- **Continuous Validation:** Unlike authentication, which is agreed upon once, identities and transactions are cryptographically validated every time access is desired.
- **Least-Privilege Access:** Access permissions are established and enforced via blockchain-based policies, granting access solely to verified users with a predefined role that requires access.
- **Microsegmentation:** Web3 enforces decentralized security controls, ensuring that even within the same network, access is segmented based on decentralized identities and smart contracts rather than traditional IP-based models.

(We will provide more details on how Web3 enforces these principles later in this chapter.)

Real-world implementations of Web3 security models include

- **Ethereum Name Service (ENS) and DID:** ENS allows users to establish blockchain-based identities that are used across decentralized applications (dApps), ensuring identity persistence without central control.
- **IBM and Hyperledger for Secure Transactions:** IBM's identity verification solutions form the bedrock for implementing secure,

decentralized authentication and preventing identity fraud in enterprise settings.

- **Decentralized Finance (DeFi) Smart Contracts:** Smart contracts govern all financial transactions in decentralized finance without intermediaries and show that smart contracts can indeed automate the tamper-proof enforcement of securities in highly sensitive environments.

This architecture is a theoretical implementation concept that highlights the possible integration between Web3 technologies with Kubernetes security. The new organization proposing to follow this will have to put in massive additional engineering effort, security audits, performance optimizations, and resilience engineering. In reality, such an implementation would have to be catered to according to organizational requirements, regulatory compliance needs, and technical capabilities. Most importantly, Web3 is, by default, a moving target, and so are cloud-native technologies—implementation details may become outdated as standards and best practices change over time.

How Decentralized Identity (DID) Works

Before we tackle how decentralized identity increases security, it is important to cover what DID is and how it works. DID is a self-sovereign identity model where you can authenticate users and applications without a central authority. Traditional identity providers (IDPs), such as Google, Microsoft, or AWS, introduce single points of failure where, if one gets compromised, millions of accounts could become compromised as well. DID eliminates this risk by

- Using cryptographic signatures on the blockchain to prove ownership of an identity

- Providing decentralized storage of identity credentials rather than centralized in a single database

- Providing verifiable credentials that can be used across platforms without revealing unnecessary user data

Rather than logging in to a system with a username and password managed by a central service, in a DID system, users cryptographically sign a challenge with their private key. The verifying party can then confirm the signature on the blockchain to ensure its authenticity. This system removes federated identity dependencies and enables self-sovereign identity management, where users control their own credentials instead of relying on third parties.

DID is a natural fit for zero trust because it enforces identity verification on every access request, ensuring users are authenticated whenever needed. It also enables fine-grained access control, where permissions are dynamically assigned based on a user's role, mission, and context of data access. This approach helps mitigate the risks of centralized identity repositories, lowering the attack surface and reducing breach risks related to compromised identities.

Hybrid Cloud Challenges and Web3's Role in Policy Enforcement

One of the biggest problems arising from hybrid cloud environments is ensuring that security policies stay consistent—across on-premises, private, and multicloud deployments. Conventional identity and access management (IAM) tools (such as Active Directory, AWS IAM, and Azure AD) tie up in identity silos, creating fragmented access controls that increase the risk of misconfigurations and expand security gaps.

Web3 technologies—particularly DID and blockchain-based policy enforcement—provide universal and permanent access control between clouds.

Web3 solutions store policies on an unchangeable blockchain and use smart contracts to dynamically enforce segmentation rules, which ensures consistent, real-time access enforcement across hybrid infrastructures. Doing so removes implicit trust, lowers attack surfaces, and increases policy transparency, regardless of where a workload is located, which further fortifies zero trust.

Decentralized Identity: A Cornerstone of Web3 Security

Decentralized identity is a foundational Web3 security model, providing users and applications a decentralized authentication system that does not require a central authority. While traditional IDPs introduce single points of failure, DID-based identity verification is secured with cryptographic signatures on the blockchain. This allows identity verification without exposing sensitive identity data while guaranteeing self-sovereign entity authentication.

DID naturally reinforces zero trust security by enforcing

- Continuous identity verification, ensuring every access request is authenticated in real time
- Fine-grained access policies, dynamically managing permissions based on role, workload context, and environment
- Elimination of centralized identity repositories, significantly reducing identity-based attack vectors

This use of DID effectively allows organizations to safeguard against attacks based on credentials and identity, which ensures that only certified, verified users can access workload segments. As a result, DID could become an essential element for access security in hybrid and multicloud environments, providing identity management that is not only strong and scalable but one that does not depend on traditional IAM systems.

Ways to Enhance Security with Blockchain’s Immutability

The best feature of blockchain technology in Web3 is its immutability. Blockchain offers the perfect environment for trust that otherwise may not be available between parties: Policies stored on a blockchain can never be tampered with. This feature is especially useful in cases where permissionless collaboration is needed. For example, organizations can create immutable smart contracts to deploy security policies at runtime. Smart contracts have no external interference that binds the agreed-upon terms that really make for better security assurances over disputes in multilateral relationships.

Security in Web3 and Smart Contracts

By decentralizing key building blocks (for example, identity management and transaction verification in cloud-native environments), Web3 technologies can also yield robust security gains. An obvious example of this is the use of blockchains to secure smart contracts: These are contracts, self-executing, self-enforcing, and requiring no referee.

On the flip side, smart contracts have their own security problems such as possible coding errors or gaps that hackers can exploit. Rolling drafts and formal verification are key to the development process because a security issue in a smart contract can lead to permanent loss of funds or interruption of critical processes.

In other words, you can learn a great deal about the importance of immutability and a trustless environment for cloud-native application security directly from smart contracts. In a decentralized environment, once a transaction or a contract is verified, it cannot be changed without the consent of the rest of the network. Zero trust dictates that nothing in cloud-native architectures should be inherently trusted. Every component—whether an application, user, or service itself—has to constantly validate its authenticity prior to access being authorized.

Blockchain for Immutable Access Logs

Blockchain can help ensure tamper-proof, unalterable access logs, enhancing segmentation and security. By leveraging blockchain-based records—which are inherently immutable—organizations can improve compliance and auditability, tracking every access attempt to segmented environments with full transparency.

Think of an entity using segmentation policy management tools alongside blockchain. In this system, each policy application and every access attempt is recorded on the blockchain, forming a secure and verifiable audit trail. This guarantees compliance, making audits seamless.

- Timestamp
- User’s DID

- Resource accessed
- Policy applied
- Access decision

In the case of access logs, most logging systems store logs in centralized databases, which subjects them to tampering or deletion. On the other hand, logs are immutable in a blockchain-based logging system so that, after the fact, they cannot be changed, guaranteeing auditability of the segmentation policy and that violations can be tracked without guesswork.

Web3's Value Beyond Blockchain: A New Security Paradigm

While blockchain and smart contracts remain the crown jewel of Web3 technologies, the real innovation spans far beyond, disrupting the very foundation of cloud-native security with new models of decentralized trust. Web3 does the same for data by removing central points of failure via peer-to-peer verification for a new approach to data integrity, identity management, and segmentation in cloud and hybrid environments.

Of the many features of the Web3 platform, one of the most powerful is decentralized identity (DID), which allows users and applications to verify each other without a central authority. DID provides a viable self-sovereign identity and cryptographic authentication framework across distributed networks, as opposed to traditional IAM solutions, which not only promote vendor lock-in but also create security bottlenecks that often lead to major breaches.

For example, larger enterprises with a centralized IAM such as Active Directory or AWS IAM often see this as a problem, where a single compromised credential or misconfiguration can result in cross-cloud environments being accessed. One clear example is the Capital One breach in 2019, when a misconfigured AWS IAM role was exploited by an attacker to access customer records of over a hundred million individuals. By contrast, DID eliminates this single point of failure by allowing each identity to be verified on a blockchain, ensuring distributed trust across multiple cloud providers. In multicloud federated authentication, this decentralized model can be extremely useful because various cloud providers can authenticate identities without relying on a single centralized IAM system.

Also, the immutability provided by Web3 to blockchains strengthens zero trust security because it makes sure access logs and security policies are tamper resistant and auditable. This echoes modern IT approaches such as Secure Access Service Edge (SASE) and edge computing, where nodes that implement security rules are distributed rather than being in a centralized data center. Web3-powered DID and decentralized access policies reduce the number of requests that require authorization from the security gateway at once and evaluate each data flow on its own rather than thousands of parallel requests from remote offices, IoT devices, devices ships, and cloud services. In turn, this establishes a powerful and scalable segmentation framework through which organizations can implement zero trust across geospatially separated cloud and edge networks.

The "never trust, always verify" principle permeates every layer of Web3, which provides a solid foundation for securing next-generation IT architectures—enhancing hybrid cloud security, empowering decentralized trust models, and solidifying zero trust at scale.

DID in Cloud-Native and Hybrid Environments

The integration of decentralized identity addresses many problems in cloud-native ecosystems while bringing transformative benefits. Specifically, conventional security techniques tend to require centralized policy enforcement, which can be cumbersome in multicloud and on-premises deployments. DID techniques make it possible to have decentralized, consistent enforcement of the policies applied across environments, reducing the attack surface, maintaining continuous compliance, and improving the confidence in the integrity of access controls.

DID improves the privacy of pseudonymous interactions with sensitive data. For instance, healthcare entities hosting electronic health records (EHR) can authenticate users to the system through DIDs without revealing personally identifiable information (PII). This approach improves sensitive data security while providing regulatory compliance such as GDPR, HIPAA, and many other frameworks.

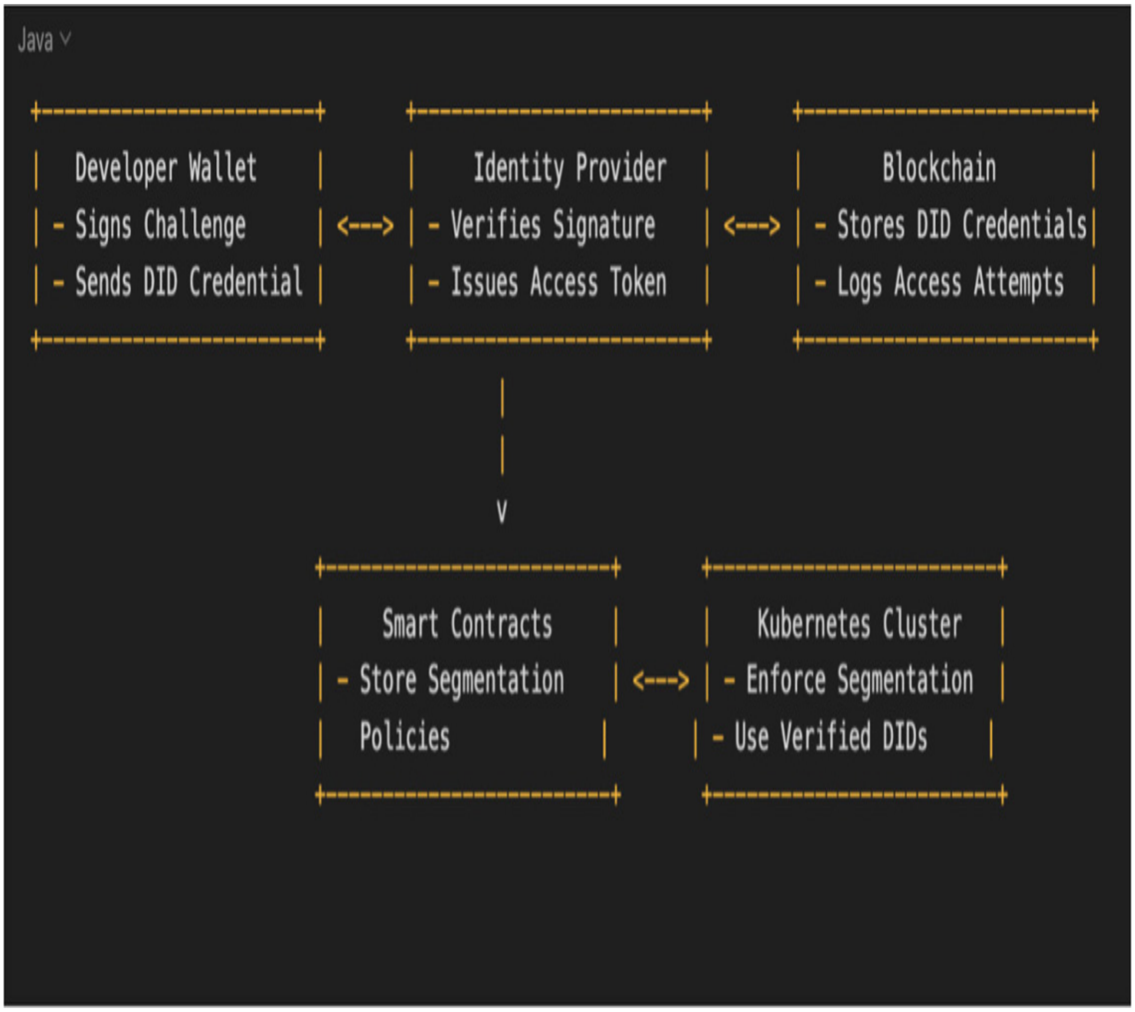
As IT evolves, Web3-backed segmentation strategies will be instrumental in securing cloud-native, multicloud, and edge environments.

Real-World Applications of DID and Zero Trust

ABC Corporation has implemented DID to secure its hybrid cloud. ABC removes dependence on a centralized IAM ecosystem that would reduce IAM-related breaches by making use of decentralized wallets to store the credentials. This approach not only streamlines operations, strengthens compliance, and enhances security by applying immutable access control to its entire multicloud infrastructure, but it also enables organizations to retain user sovereignty through its blockchain policies.

Workflow: Kubernetes Access Using DID and Smart Contracts

Figure 17-11 illustrates *authorization flow*, where a cloud-native developer at ABC Corp uses a DID credential to prove their role and organizational association when accessing a Kubernetes namespace.



```
+----------------------+    +----------------------+    +----------------------+
|   Developer Wallet   |    |   Identity Provider  |    |      Blockchain      |
| - Signs Challenge    |<-->| - Verifies Signature |<-->| - Stores DID Credentials|
| - Sends DID Credential|   | - Issues Access Token|    | - Logs Access Attempts |
+----------------------+    +----------------------+    +----------------------+
                                   |
                                   |
                                   v
                    +----------------------+    +----------------------+
                    |   Smart Contracts    |    |  Kubernetes Cluster  |
                    | - Store Segmentation |<-->| - Enforce Segmentation |
                    |   Policies          |    | - Use Verified DIDs  |
                    +----------------------+    +----------------------+
```

Figure 17-11 *Workflow for Kubernetes Access Using DID and Smart Contracts*

This workflow allows verification and zero trust enforcement directly on the chain rather than through passwords or centralized authentication. Let's walk through this in detail. A step-by-step authorization flow might look like this:

1. **User Requests Access**

   a. A developer at ABC Corp attempts to access a Kubernetes namespace or deploy a pod.

2. **DID-Based Authentication and Authorization**

   a. Kubernetes delegates authorization decisions to an external Web3-based policy engine via a validating admission controller.

   b. The policy engine retrieves DID credentials and queries the blockchain for real-time access verification.

3. **Smart Contract Evaluation**

   a. The policy engine queries the blockchain to

      - Validate the user’s DID credentials

      - Check role-based policies associated with the requested Kubernetes resource

      - Apply additional constraints (such as time-restricted access, least-privilege enforcement)

4. **Authorization Decision**

   a. Based on smart contract logic, the external policy engine returns an allow/deny verdict.

   b. Kubernetes enforces the decision accordingly.

5. **Kubernetes Enforces Access Control**

   a. If access is granted, Kubernetes applies segmentation policies through

      - Role-based access control (RBAC)

      - Network policies

      - Admission controls

Coding Mistakes and Exploits

Because of the immutable nature of smart contracts, any coding errors, logical errors, or unverified dependencies will create critical security holes when deployed. Flaws in smart contract logic are exploitable vulnerabilities that may allow unauthorized access, privilege escalation, data corruption, or financial loss. Further, vulnerabilities in badly coded smart contracts can lead to attack vectors such as denial of service (DoS), frozen assets, or accidental privilege settings.

One of the most infamous of these types of exploits is the DAO Hack in 2016, when an Ethereum smart contract with an exploitable reentrancy vulnerability was exploited and drained of more than $60 million of ETH (ether). It was a bug in the contract processing calls to a recursive function, allowing the attacker to call it multiple times before updating the balance and to take out money over and over. This exploit highlighted the dangerous side of immutable smart contracts, namely that a single line of code can lead to large monetary and security ramifications.

Rolling drafts and formal verification reduce these risks by requiring smart contracts to be tested and validated using advanced mathematical techniques before they are deployed. Chain governance mechanisms ensure an additional level of security by allowing contract updates or revocations to be performed in a safe and controlled manner via a consensus-based approach. Automation of security audits and penetration testing allows recognition of vulnerabilities that will be exposed in the production environment. Contract design with a least-privilege process allows you to limit potential attack vectors by giving contracts only the permission required for them to execute their intended functions, hence minimizing attack surfaces and potential exploits.

Latency Concerns

Frequently querying the blockchain for every authorization request might introduce additional delays, and even more when being dependent on a middleware component, which may not be sufficient in terms of added latency for the cloud-native applications because the time taken to get and validate those DID credentials and policies from the blockchain is high. To address this situation, organizations can implement off-chain caches to store frequently used policies and verification results, reducing the need for repeated on-chain queries. Moreover, they can also utilize Layer 2 scaling solutions such as Polygon or the Optimistic Rollups to increase the speed of blockchain transactions while keeping the costs low.

Layer 2 scaling methods are technologies built on top of an existing blockchain (Layer 1) designed to help with speed and congestion in transactions. Layer 2 solutions process multiple transactions in batches or process them off-chain, and update the main chain (in this case Ethereum Layer 1) with a single validity proof.

- **Polygon:** This Layer 2 scaling solution processes transactions using sidechains, allowing it to do so at a higher speed and cheaper than the Ethereum main network. This allows Ethereum-compatible applications to reach high throughput without compromising on security.

- **Optimistic Rollups:** This Layer 2 technology executes the transactions off-chain and only submits the final state back to the main chain. By default, this technology assumes that transactions are valid by default, only performing additional verification if fraud is suspected, leading to significantly faster and cheaper transactions.

Implementations must make use of a multilevel caching strategy to address the performance impact of blockchain queries. Examples include caching validated DIDs and prior authentication results in a distributed cache like Redis. Additionally, organizations can implement local caches within the Web3 Policy Engine for frequently accessed policies and use event subscriptions to maintain an eventually consistent local state. It may also make sense at a given moment, for a given organization, to use Layer 2 blockchain solutions or even sidechains, which then provide lower latency and lower cost but with security guarantees preserved by periodically committing states to the main chain.

Smart Contract Revocation Challenges

Smart contracts are immutable, transparent, and are an essential part of most blockchains, However, revoking permissions by updating access policies is not instantaneous. Even if a contract allows updates, modifying on-chain permissions requires a blockchain transaction, which must be mined and confirmed, causing delays. Additionally, off-chain systems that rely on blockchain state may take time to reflect the changes. To mitigate this problem, organizations can add off-chain revocation lists alongside blockchain policies and implement short-lived credentials, such as tokens that expire every 24 hours, ensuring access is validated regularly.

This implementation will need solid error management at various levels. These critical issues include unavailability of blockchain networks, failures in the execution of smart contracts, and verification of DIDs. This means implementing circuit breakers that can fall back to secondary methods of authentication in the event of a blockchain network outage, mechanisms to retry failed transactions with exponential backoff, and dead-man switches that will keep the system from locking out absolutely. Moreover, system monitoring and alerting must also be put in place to detect authentication and authorization failures.

Kubernetes Webhook Security Risks

Although Kubernetes admission controllers and webhooks are incredible tools for binding external authorization systems, they also become potential attack vectors. In cases where a webhook or external policy engine is compromised, the policy enforcement can be caused by malicious actors. Organizations can help prevent this risk by ensuring only Web3 verification services that implement mTLS authentication on webhook communications are allowed and setting fail-safe defaults so that if the Web3 verification service goes down, access would be denied until the service comes back online.

Blockchain Cost and Scalability

Running smart contracts for policy evaluations mandates transaction fees (*gas costs*, which is the unit that measures computational work and is paid in the blockchain's native cryptocurrency), which can become prohibitively expensive in high-demand environments. This limitation also impacts scalability when dealing with large Kubernetes clusters or frequent policy checks. Although querying a smart contract via a Remote Procedure Call (RPC) endpoint is free, certain validations incur gas costs. These include verifying execution as a specific wallet (such as signing messages or transactions), modifying the contract's state or code, deploying new contracts, or invoking state-changing functions (e.g., marking a user as a developer or revoking access).

To mitigate costs, organizations can use fee-less or low-cost blockchain networks, like Hedera or Solana. Additionally, batching multiple verification requests into a single transaction can optimize resource usage and reduce fees.

Integrating Kubernetes with Blockchain for Smart Contract–Based Access Control

Kubernetes itself does not directly communicate with blockchains but can delegate authorization decisions to a third-party Web3-based policy engine. This enables DID-based authentication, blockchain-enforced policies, and dynamic segmentation.

Decentralized Identity Authentication in Kubernetes with Keycloak SPI

To integrate DID authentication, Kubernetes uses Keycloak's Authentication Service Provider Interface (SPI) to extend authentication to Web3 wallet-based logins instead of traditional credentials.

Keycloak's SPI allows a Web3 wallet to sign a cryptographic challenge instead of requiring a username-password login. This signed challenge is verified against the blockchain, ensuring that only the rightful owner of a DID can authenticate. If the signature is valid, Keycloak issues an OIDC token embedding blockchain-verified credentials, which Kubernetes uses for access control enforcement.

OpenID Connect (OIDC) is an authentication protocol that uses OAuth 2.0 to provide identity verification by issuing identity tokens instead of using credentials. This protocol enables applications to authenticate users without the need to reveal passwords, and instead, utilize identity providers (IdPs) to create and pass JSON Web Tokens (JWT) containing claims to a verifiable identity. Because of its ability to implement federated authentication across hybrid and multicloud, OIDC has become a mainstay for cloud-native security and is widely used.

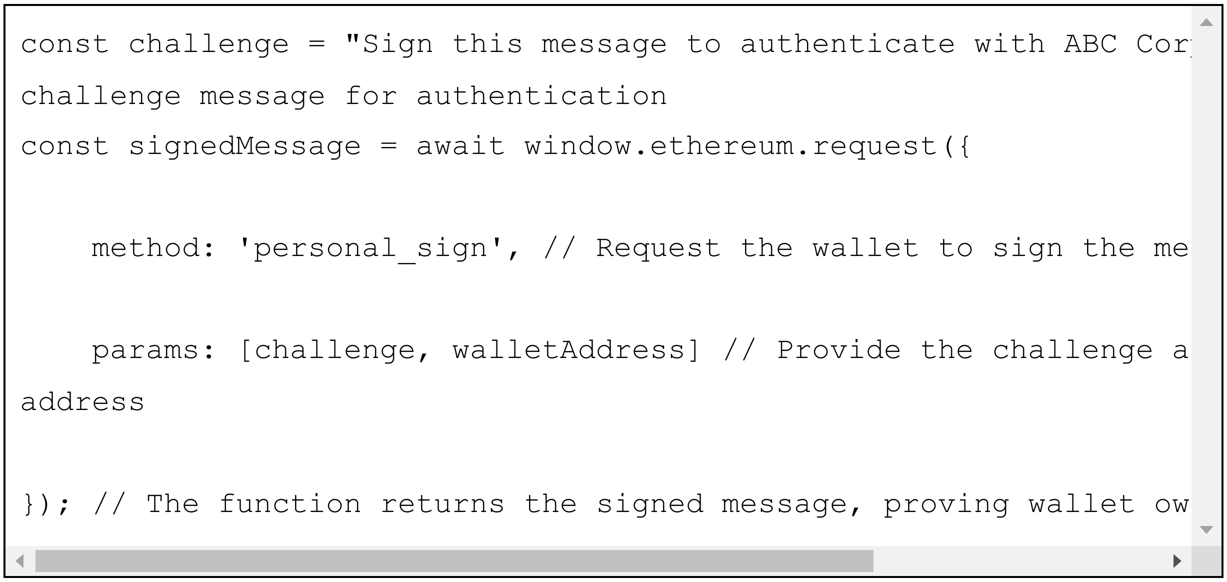
Authentication Flow Implementation Details for DID and Smart Contracts

A user begins authentication by signing a challenge with a Web3 wallet. The signed challenge is then sent to Keycloak’s SPI for verification. The SPI extracts the wallet address from the signature and queries an external DID resolver or smart contract to validate the user’s identity. If successful, Keycloak issues an OIDC token containing verified claims. This token is then used to request access in Kubernetes, where a validating admission controller consults a smart contract to determine if the request is authorized based on DID credentials and policy enforcement.

User Initiates Authentication (Signing Challenge with Web3 Wallet)

A developer at ABC Corp tries to log in to a cloud-native system using a Web3 wallet (such as Keplr, MetaMask, or HyperSign). Instead of the developer entering a username and password, the Web3 wallet signs a challenge message with the user’s private key. Example 17-1 shows a JavaScript snippet that demonstrates how a Web3 wallet signs an authentication challenge before sending it to Keycloak’s authentication SPI for verification.

Example 17-1 *Signing an Authentication Challenge Using a Web3 Wallet*



```
const challenge = "Sign this message to authenticate with ABC Cor
challenge message for authentication
const signedMessage = await window.ethereum.request({

    method: 'personal_sign', // Request the wallet to sign the me

    params: [challenge, walletAddress] // Provide the challenge a
address

}); // The function returns the signed message, proving wallet ow
```

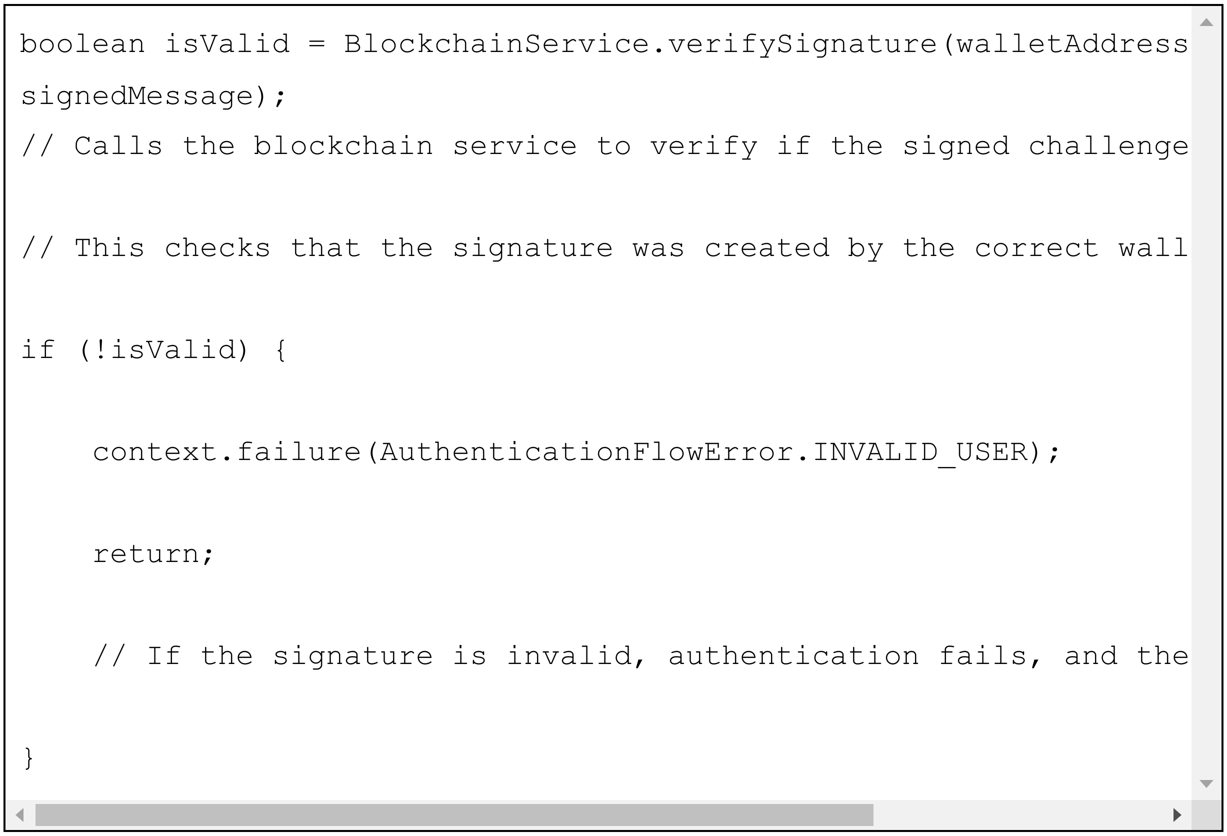
This signed challenge is sent to Keycloak’s authentication SPI.

Keycloak SPI Validates Signature and DID

The SPI in Keycloak receives the signed challenge and verifies the user’s wallet address extracted from the signed message and the cryptographic validity of the signature. It then interacts with an external policy engine (or blockchain DID resolver) to confirm wallet ownership on the blockchain and check DID credentials (such as whether the user is a developer at ABC Corp). If the DID credential is valid, Keycloak proceeds to token issuance.

Keycloak validates the user’s signature by checking it against the public key stored on the blockchain, as shown in Example 17-2.

Example 17-2 *Verifying a Signed Authentication Challenge Using Blockchain*



```
boolean isValid = BlockchainService.verifySignature(walletAddress
signedMessage);
// Calls the blockchain service to verify if the signed challenge

// This checks that the signature was created by the correct wall

if (!isValid) {

    context.failure(AuthenticationFlowError.INVALID_USER);

    return;

    // If the signature is invalid, authentication fails, and the

}
```

The OIDC Token Is Issued

Once the signature is verified and the DID credential is authenticated, Keycloak issues an OIDC access token embedding the user's DID credentials (see Example 17-3).

Example 17-3 *OIDC Token Issued by Keycloak with Embedded DID Credentials*

```
{
  "sub": "did:abc:1234",
  "wallet_address": "0x1234abcd...",
  "role": "developer",
  "verified_did": true,
  "exp": 1672531200
}
```

The JWT token is cryptographically signed using Keycloak's private key, ensuring integrity and allowing Kubernetes to trust the claims.

Kubernetes Validates the OIDC Token

Kubernetes must be configured to trust Keycloak as an OIDC provider to verify authentication requests. The Kubernetes API server needs to be explicitly configured with the OIDC settings to validate tokens issued by Keycloak (see Example 17-4).

Example 17-4 *Configuring Kubernetes API Server for OIDC Authentication*

```
kube-apiserver \
  --oidc-issuer-url=https://keycloak.example.com/auth/realms/my-r
  --oidc-client-id=kubernetes \
  --oidc-username-claim=sub \
  --oidc-groups-claim=role \
  --oidc-ca-file=/etc/kubernetes/ssl/ca.pem
```

The OIDC token validation steps in Kubernetes are as follows:

1. Kubernetes fetches Keycloak's public key from the /.well-known/openid-configuration endpoint.
2. It validates the signature of the OIDC token to ensure authenticity.
3. It extracts the user's DID and role from the token claims.

If validation fails, the request is immediately denied, ensuring that unauthorized users cannot interact with the cluster.

Developer Uses Token to Access Kubernetes

Once the developer successfully authenticates using the DID-based Web3 wallet and receives an OIDC token from Keycloak, they include this token in their API requests to Kubernetes. Since Kubernetes does not natively support DID-based authentication, the process can be bridged using Kubernetes ServiceAccounts, OIDC token validation, and a validating admission controller (see Example 17-5). Kubernetes forwards each request to the validating admission controller for real-time policy evaluation before granting access.

Example 17-5 *Using kubectl with an OIDC Token*

```
kubectl --token=$(cat oidc-token.txt) get pods --namespace=stagin
```

Alternatively, the developer can configure Kubernetes to use the OIDC token automatically by updating the Kubeconfig file (Example 17-6).

Example 17-6 *Configuring Kubeconfig for OIDC Token Authentication*

```
apiVersion: v1
kind: Config
clusters:
  - name: my-cluster
    cluster:
      server: <https://k8s-api.example.com>
contexts:
  - name: my-context
    context:
      cluster: my-cluster
      user: developer
users:
  - name: developer
    user:
      auth-provider:
        name: oidc
        config:
          idp-issuer-url: <https://keycloak.example.com/auth/real
          client-id: kubernetes
          client-secret: my-secret
          id-token: <INSERT-OIDC-TOKEN-HERE>
          refresh-token: <INSERT-REFRESH-TOKEN-HERE>
current-context: my-context
```

Now, every kubectl request automatically sends the OIDC token to Kubernetes, eliminating the need for manually passing the token with every command.

Kubernetes Enforces Access Using Smart Contracts Authorization

When a developer attempts to deploy a pod in Kubernetes, the request is intercepted by the validating admission controller, which queries the Web3 policy engine. The policy engine interacts with a smart contract to verify if the developer’s DID credentials meet the required role, whether access is restricted by time or context, and if additional multifactor authentication is needed. If authorization is granted, Kubernetes enforces segmentation and applies relevant security policies, ensuring that access is strictly controlled.

As a result, the developer never enters a password, and Kubernetes only grants access based on blockchain-verified credentials. Every access decision is logged immutably to maintain auditability and compliance. This guarantees zero trust enforcement, where every request is verified on-chain before approval, eliminating unauthorized access and ensuring cryptographic security at every level.

External Admission Controller (Validating Webhook)

A validating admission controller enforces blockchain-based access control policies in Kubernetes by intercepting requests and submitting them to an external Web3 policy engine, which evaluates smart contract logic before Kubernetes approves or denies access. This ensures that all access requests comply with DID verification and blockchain-enforced policies.

To implement this, a validating webhook is configured to check specific resources, such as Pods and Namespaces, during creation or updates. The webhook submits these requests to a policy engine running inside Kubernetes. Example 17-7 shows a sample webhook configuration.

Example 17-7 *Validating Webhook Configuration for Blockchain-Based Access Control*

```
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration
metadata:
  name: blockchain-policy-webhook
webhooks:
  - name: blockchain-policy.abc-corp.com
    clientConfig:
      service:
        name: blockchain-policy-service
        namespace: default
        path: "/validate"
      caBundle: <base64-encoded-cert>
    rules:
      - operations: ["CREATE", "UPDATE"]
```

```
        apiGroups: [""]
        apiVersions: ["v1"]
        resources: ["pods", "namespaces"]
    admissionReviewVersions: ["v1"]
```

To handle these requests, a Web3 policy engine is deployed as a containerized microservice inside Kubernetes. This middleware interacts with three key components: the Kubernetes Admission Controller, which receives access requests; the Blockchain Network, which executes smart contract-based policy validation; and the DID Resolver, which verifies decentralized identities before access is granted.

The Kubernetes integration relies on standard OIDC authentication flows but requires additional components for Web3 integration. The Web3 policy engine is deployed as a Kubernetes *deployment*, ensuring decentralized access control enforcement, and should be implemented as a highly available service. Example 17-8 shows a Kubernetes deployment configuration.

Example 17-8 *Deploying the Web3 Policy Engine in Kubernetes*

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web3-policy-engine
  namespace: security
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      app: web3-policy-engine
  template:
    metadata:
      labels:
        app: web3-policy-engine
    spec:
      containers:
        - name: policy-engine
          image: myregistry/web3-policy-engine:v1
          ports:
            - containerPort: 5000
          livenessProbe:
            httpGet:
              path: /health
              port: 5000
            initialDelaySeconds: 15
            periodSeconds: 20
          readinessProbe:
            httpGet:
              path: /ready
              port: 5000
            initialDelaySeconds: 5
            periodSeconds: 10
          resources:
            limits:
              cpu: "1"
              memory: "1Gi"
            requests:
              cpu: "500m"
              memory: "512Mi"
          env:
            - name: BLOCKCHAIN_NODE_URL
              value: "<https://eth-node.example.com>"
            - name: SMART_CONTRACT_ADDRESS
              value: "0x123456789abcdef"
            - name: DID_RESOLVER_URL
```

```
              value: "<https://did-resolver.example.com>"
            - name: CACHE_TTL
              value: "300"
            - name: MAX_RETRIES
              value: "3"
            - name: BLOCKCHAIN_FALLBACK_ENABLED
              value: "true"
```

This enhanced deployment configuration includes health monitoring through readiness and liveness probes, resource limits and requests for proper scaling, and additional environment variables for caching and fallback mechanisms. The RollingUpdate strategy ensures zero-downtime deployments, while resource constraints prevent the policy engine from consuming excessive cluster resources.

By leveraging blockchain-backed policy validation, this middleware eliminates centralized IAM dependencies, making Kubernetes access control tamper-proof, fully auditable, and cryptographically verifiable. Every access request is validated against on-chain policies, ensuring zero trust security principles while enhancing compliance and decentralized authentication.

Kubernetes Enforces DID-Based Segmentation

Once access is authorized via smart contracts, Kubernetes enforces additional DID-based segmentation policies, dynamically defining access rules to ensure that only verified users can interact with specific cloud-native resources. A basic network policy for DID-based access control might look like that shown in Example 17-9.

Example 17-9 *Network Policy for DID-Based Access Control in Kubernetes*

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: staging-segmentation
  namespace: staging
spec:
  podSelector:
    matchLabels:
      app: web
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              verified-did: "true"
```

This policy automatically restricts access based on blockchain-verified credentials.

Smart Contracts for Policy Enforcement

Smart contracts enforce role-based, time-based, and privilege-aware access control policies, ensuring that only authorized users can interact with Kubernetes resources. These contracts implement user role verification, ensuring that only individuals with the correct role (for example, developer) are granted access. Additionally, they support context-aware access, applying time-based or geolocation-based conditions to further restrict permissions. By implementing least-privilege enforcement, smart contracts limit access strictly to predefined Kubernetes namespaces, reducing the risk of unauthorized resource manipulation while maintaining a secure, decentralized, and auditable access control mechanism.

Solidity Contract Example

Example 17-10 shows a partial smart contract implementing Kubernetes access control logic.

Example 17-10 *Smart Contract Code with Inline Explanations*

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract KubernetesAccessControl {
    // Mapping to store namespace permissions for each user.
    // Example: allowedNamespaces[userAddress]["staging"] = true
access the "staging" namespace.
    mapping(address => mapping(bytes32 => bool)) private allowedN

    // Mapping to store developer roles. Only users marked as "tr
developers.
    mapping(address => bool) private developers;

    // Allowed time window for access (e.g., working hours).
    uint256 public allowedTimeWindowStart;
    uint256 public allowedTimeWindowEnd;

    // Address of the contract owner (has permission to assign ro
    address public owner;

    // Events to log role assignments, permission changes, and ac
    event DeveloperAssigned(address indexed user);
    event DeveloperRevoked(address indexed user);
    event NamespacePermissionGranted(address indexed user, bytes3
    event NamespacePermissionRevoked(address indexed user, bytes3
    event AccessChecked(address indexed user, bytes32 resource, b

    // Modifier to restrict certain functions to the contract own
    modifier onlyOwner() {
        require(msg.sender == owner, "Not authorized");
        _;
    }
    // Constructor: Sets the contract owner and defines the allow
    constructor(uint256 startTime, uint256 endTime) {
        owner = msg.sender;
        allowedTimeWindowStart = startTime;
        allowedTimeWindowEnd = endTime;
    }

    // Function to assign the "developer" role to a user (only co
this).
    function assignDeveloperRole(address user) public onlyOwner {
        developers[user] = true;
        emit DeveloperAssigned(user);
    }

    // Function to revoke the "developer" role from a user.
    function revokeDeveloperRole(address user) public onlyOwner {
        developers[user] = false;
        emit DeveloperRevoked(user);
    }

    // Function to grant a user permission to access a specific K
    function setNamespacePermission(address user, bytes32 resourc
        allowedNamespaces[user][resource] = true;
        emit NamespacePermissionGranted(user, resource);
    }

    // Function to revoke a user’s permission to access a specifi
    function revokeNamespacePermission(address user, bytes32 reso
{
        allowedNamespaces[user][resource] = false;
        emit NamespacePermissionRevoked(user, resource);
    }
    // Function that Kubernetes queries to verify access.
    function verifyAccess(address user, bytes32 resource) public 
        // Check if the user has the developer role.
        if (!developers[user]) {
            emit AccessChecked(user, resource, false);
            return false;
        }

        // Ensure the request is made within the allowed time win
        if (block.timestamp < allowedTimeWindowStart || block.tim
allowedTimeWindowEnd) {
            emit AccessChecked(user, resource, false);
            return false;
        }

        // Check if the user has permission to access the request
        if (!allowedNamespaces[user][resource]) {
            emit AccessChecked(user, resource, false);
            return false;
        }

        // If all conditions are met, access is granted.
        emit AccessChecked(user, resource, true);
        return true;
    }
}
```

Overall, the smart contract implementation provided in this example serves as a working implementation but will need to be hardened for production use. Some critical security features may involve using reentrancy guards for all state-modifying functions, proper access control mechanisms beyond basic ownership, DID revocation tracking, emergency pause functionality, and comprehensive event emission for all state changes. The organizations must also put in place an appropriate upgrade mechanism to patch vulnerabilities and conduct detailed security audits of the smart contract code before going live.

Audit Logging: Immutable Developer Access Logs

Once access is granted (or denied), a blockchain log entry is generated for auditability and compliance (see Example 17-11).

Example 17-11 *Blockchain-Based Immutable Audit Log for Kubernetes Access Control*

```
{
  "timestamp": "2025-01-27T10:15:00Z",
  "user_did": "did:abc-corp:dev123",
  "resource": "k8s/staging-app",
  "action": "read",
  "policy_id": "policy-5678",
  "decision": "allow"
}
```

This entry ensures security, decentralization, and tamper-proof policy enforcement in Kubernetes by creating an immutable audit trail that prevents unauthorized modifications to logs. It also ensures regulatory compliance with frameworks such as GDPR, HIPAA, and SOC 2, aligning with defined organizational compliance policies. Additionally, it provides a reliable record for forensic analysis, supporting incident investigations by maintaining a transparent and verifiable history of access attempts. This guarantees that every access event is securely logged, eliminating security blind spots and reinforcing a zero trust security model.

Benefits of This Approach

Because it is no longer depending on centralized IAM systems, ABC Corporation mitigates both credential theft risks and privilege escalation threats. Every access request is checked against blockchain-based policies in real time, which supports zero trust by preventing unauthorized access and privilege escalation within its Kubernetes environments.

The approach supports dynamic segmentation—allowing only those developers with appropriately verified DID credentials to interact with sensitive workloads. In contrast to static role-based access controls, these policies are dynamic, taking available context into account, and continually assessed; they automatically adapt to changing security needs on the fly—without manual intervention.

The integration with Kubernetes allows ABC to enforce fine-grained segmentation policies, making access highly dynamic and conditional. Such resilience and decentralization mean that as long as the policy is on the blockchain, ABC access control is safe because it does not rely on a single IAM provider, which can be hacked more easily.

In addition, every authorization decision and access log is permanently stored on-chain, and thus provides an irrefutable chain of auditable data for regulatory compliance and forensic analysis. Consequently, this architecture positions security posture in a much more optimized manner across multicloud, hybrid, as well as edge computing environments while maintaining auditability, transparency, and more robust zero trust enforcement.

Summary

Segmentation relies on an organization’s flexibility to adopt multicloud architectures and edge computing or to address the changing threat landscape, and is an integral part of any cloud-native application security approach. Utilizing tools such as the ones mentioned in this chapter or similar frameworks, while pushing segmentation to the edge, allows organizations to ensure that their policy enforcement capabilities are consistent, scalable, and resilient. This ensures segmentation will continue to play a key role in the overall cloud-native application security initiatives, in a scalable and compliant way, to prevent security threats in tomorrow’s dynamic IT environments.

References

1. “No More Chewy Centers: The Zero Trust Model of Information Security”: https://www.forrester.com/report/No-More-Chewy-Centers-The-Zero-Trust-Model-Of-Information-Security/RES56682

2. “Cisco Hybrid Multi-Cloud Networking Design Guide”: https://www.cisco.com/c/en/us/td/docs/dcn/whitepapers/cisco-hybrid-multicloud-networking-design-guide.xhtml

3. “Cisco Application Centric Infrastructure (ACI) Design Guide”: https://www.cisco.com/c/en/us/td/docs/dcn/whitepapers/cisco-application-centric-infrastructure-design-guide.xhtml

Chapter 18. Using Common Policy to Enforce Security

In this chapter, you will learn about the following:

- Security policies
- Cloud security policies' strategic frameworks
- Integration of security policies within SDLC, SDL, and SSDLC frameworks
- The OWASP SAMM framework-agnostic model
- Policy enforcement mechanisms
- Data protection and privacy policies
- The taxonomy of SDLC, highlighting common vulnerability causes
- Unified monitoring, logging, and auditing policies
- Incident response frameworks and automated remediation tools
- Key tools and technologies, including Cisco's Security Cloud

Introduction to Security Policies

Among the most appreciated advantages of cloud computing are its flexibility and your ability to work from anywhere. However, it does come with a higher risk: Unsecured networks and devices or the lack of access restrictions enables hackers to intrude into an organization's cloud environment and steal sensitive data. Since the massive shift to remote work in recent years and the ever-increasing number of remote and frontline workers, businesses have become more concerned about maintaining the security of their cloud environments. Developing a robust cloud security policy has never been more crucial.

What Is a Cloud Security Policy?

A cloud security policy is more than just a set of rules; it’s a strategic framework designed to protect cloud-based assets and systems. These policies may differ from one organization to another, but they all share common goals: reducing risks, meeting compliance requirements, and preparing for potential security threats.

For any company, security policies are the center of any robust cybersecurity program. They help guide the process of securing sensitive data in a way that is compliant with laws and industry standards. But that plan becomes more complicated in a hybrid cloud and on-premises environment where security policies start to diverge.

These policies need to strike a balance between being comprehensive enough to cover diverse systems and flexible enough to adapt to different environments. The key lies in building policies on a foundation of clear principles and actionable best practices—ones that don’t just exist on paper but guide real-world decisions and enforcement.

Principles of Effective Policy Management

Effective policy management constitutes governance, risk management, and compliance at all levels of the organization. Policies must link to the context, goals, values, and strategies of the organization, in accordance with the company’s risk appetite and operational model. Integration of monitoring into business operations is necessary, along with centralized policy oversight and acceptance of policy responsibilities in affected operations. Policy management is about human behavior and workplace culture. Whether employee, client, or third-party relationships, it should be people-centered.

High-performing policy management must be all of these things—effective, resilient, efficient, and agile. Consequently, developing standardized policies and procedures will lead to better understanding and create an auditable trail to defend the organization, while inter-departmental collaboration facilitates participation in policy management and individual policy writing.

Official policies should be accessible at all levels of the organization, ensuring a comprehensive perspective. Additionally, policies should be maintained dynamically, using clear and engaging language, to support continuous improvement as business objectives, operations, and risk profiles evolve.

Designing Common Security Policies

Common security policies can and should be designed in such a way that you can define a standard framework that can be uniformly applied to every environment. This balance between granularity and manageability is required to ensure that policies will be detailed enough to meet specific security goals, while not too complicated for a tenant to implement and maintain. The policies are normalized over different environments to simplify them, and the control becomes the same, which reduces all complexities; hence, the procedure will be secure, and the posture will automatically get stronger.

Unifying Security Policy Frameworks

The initial approach to building a cohesive security policy framework is recognizing the needs of each environment—on-premises, cloud, and hybrid. A consolidated framework should define high-level guidelines that could form a common foundation across all environments, while specific needs should be defined based on the unique environment in the context of governance.

Let’s continue our discussion with our fictional company, ABC Corporation. ABC may define a set of common policies covering user authentication, data encryption, and access controls that apply across all its systems. Establishing a unified security policy framework requires first understanding these varied requirements.

Balancing Granularity and Manageability

Policy design requires a well-thought-out balance between granularity of your policies and manageability, because the more granularity that you impose on your policies, the less manageable they will become. Granular policies provide detailed control over security configuration, but excessively fine granularity makes enforcement complex and prone to variability.

In network access control, for example, a granular policy may say that only a particular set of IP addresses can access a few certain critical servers. If the policy is very comprehensive, however, and defines unique rules for every server, it is largely unmanageable. A more balanced approach is setting a base network access policy (for example, all critical servers must block access to known IP addresses but provide exceptions only when required).

Standardizing Policies Across Diverse Environments

Cross-environment enforcement is about making policies that can be enforced without attention to specific technology (the same policy can be deployed everywhere). To do this, the policy goals need to be abstracted from the underlying implementation details.

For example, the data backup policy for ABC Corp may require that all critical data be backed up on a daily basis and stored in a secured off-site location. On-premises, this can be local backup solutions like tape drives or an on-premises backup server, and off-site it may be through a secured transport service. That same policy would leverage cloud provider backup services in the cloud, with the same backups stored in a different geographical region for regional outages.

Creating Consistency Across Environments

Policy templates and automated policy enforcement tools can help enforce policies consistently. Tools like policy management software can help to automate the deployment and enforcement of policies across various environments.

For example, ABC Corp has a policy management tool that automatically enforces courtesy security policies defined in its enterprise environment across all of its cloud and on-premises environments. This means that multifactor authentication (MFA) will be enforced for all user accounts, the same encryption standards will apply to data storage solutions in both environments, and role-based access controls (RBACs) will be set up uniformly across all systems, thus preventing unauthorized access.

Adapting to Changing Access Patterns with Common Policy

Network administrators face new challenges in the post-pandemic world, where access patterns have dramatically changed. End users now work from different places, like remote sites, head offices, campuses, and branches, so they log in to a hybrid model. Regardless of whether users, devices, or application workloads are remote or on-premises, this transition requires consistent segmentation and access baselines by the very nature of zero trust.

The solution to handle this complexity is adopting a *Common Policy framework*—a common language for access and segmentation policies in all network domains. Traditionally, these policies do not share the ability or ease of communication across domains, WAN, LAN, or cloud, so each domain has a separate structure to implement the policies. As an example, WAN policies may depend on IPs, but the data center policy is based on endpoint groups (EPGs) or endpoint security groups (ESGs), for example.

Cisco's Identity Services Engine (ISE) 3.4 utilizes Common Policy to put context information into security group tags (SGTs) that can be understood by any domain, solving the issue of lack of context between different security domains. This normalization enables network administrators to define uniform access and segmentation policies independent of the domain where they are applied. The context information—such as user identity, device type, and application workload—is created closer to the domain where it resides, whether at the access layer for users and devices or in the data center or cloud for application workloads.

With Cisco ISE as an exchange hub, the normalized context is shared across domains, enabling security administrators to enforce consistent policies. That helps end the confusion of differences in terminology and policy constructs across different network segments. As an example, in a multipod, multitenant, and multi-VRF (virtual routing and forwarding) environment Cisco ISE can naturally translate EPG and ESG to SGT, which makes it a lot easier to implement uniform policies across the segments.

Additional zero trust efforts occur with Common Policy, which collects context, saves it, and provisions it to other network domains (and controllers). Giving NetOps and SecOps teams the flexibility to pick the domain where they want policies enforced, this capability enables teams to push policies where they matter most.

Essentially, Common Policy becomes the translator for your network, making sure that every segment speaks the same language. This comprehensive approach captures all facets of the network and provides consistent policy management and security for users, devices, and application workloads, regardless of where they're located or the spanning domain. Common Policy adoption helps organizations build on their existing zero trust security posture and ensure unified access and segmentation policies across their entire network infrastructure.

Policy Enforcement Mechanisms

Policy enforcement tools use a variety of technologies to monitor and control network traffic according to security policy. In the following sections, we will touch on a few mechanisms and show an implemented example.

Firewalls and Intrusion Detection/Intrusion Prevention Systems (IDS/IPS)

You can think of a firewall as your first line of defense for preventing attacks; it essentially acts like a bouncer who stands at the door to let in people (traffic) only according to certain rules that you have defined. Intrusion detection and intrusion prevention systems (IDS/IPS) are used to detect potential threats and work to prevent breaches—a great complement with firewalls.

In this example, ABC Corp uses a Layer 7 next-generation firewall (NGFW) to examine traffic from the application down to the network. As the name implies, this is a security measure positioned in between two devices; everything that goes through is actual traffic while anything risky is discarded. At the same time, IDS/IPS detect imperfections in network behavior, such as uncharacteristic attempts to log in to a system or patterns that signal data leaving the company's control. Normally, an IPS can block detected malware traffic automatically, and an IDS sends out alerts to the organization's security team for inspection before taking any action. This two-tiered method boosts ABC Corp's ability to thwart rated and unrated threats from outside the network.

Cloud Access Security Brokers (CASBs)

Cloud Access Security Brokers (CASBs) play a critical role in cloud security. These tools provide visibility into cloud application usage and enforce policies for data security, compliance, and threat protection. For example, ABC Corp uses a CASB to monitor and regulate data transfers between its cloud applications and on-premises systems. The CASB enforces policies that prevent sensitive data from being uploaded to noncompliant cloud services. This ensures that customers' personally identifiable information (PII) is stored only on approved platforms, such as Microsoft OneDrive, where stronger security measures reduce the risk of data breaches.

The Role of CASB in Cloud Security

Cloud security becomes a major concern when organizations migrate their data and applications to the cloud. The shared responsibility guarantees between organizations and cloud providers underlie the need for enterprises to be active participants in shielding their own data security stance in any aspect of a cloud provision. Here, the most powerful solution to provide conditional cloud security is a Cloud Access Security Broker. The CASB is a security tool that sits between the on-premises infrastructure of an organization and the cloud provider’s infrastructure, serving as a gatekeeper when enforcing security policies.

A CASB provides crucial visibility and control over cloud environments, allowing organizations to manage their data and applications on vendor platforms effectively. CASBs monitor system, application, and sensitive data activity, enforcing user-based policies to prevent unauthorized access or information sharing. They secure files in motion; limit unsanctioned apps from accessing sensitive services like registration, HR, and legal systems; and ensure compliance with various regulatory requirements such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry Data Security Standard (PCI DSS).

CASBs also offer advanced threat protection by using machine learning (ML) and artificial intelligence (AI) to detect security threats in real time. They identify malicious activities, monitor irregular user accounts, and trigger actions to help security teams protect corporate systems. Furthermore, CASBs can integrate with other security tools, such as security information and event management (SIEM), data loss prevention (DLP), and security orchestration, automation, and response (SOAR) platforms, to create a comprehensive security solution. For example, SIEM integration allows security events in the cloud to be correlated with on-premises events, enabling effective security incident handling, while DLP integration helps monitor and control sensitive data within the infrastructure.

Security Orchestration, Automation, and Response (SOAR)

Security orchestration, automation, and response platforms are built to automate security incident responses through predesigned operation policies, among other capabilities aimed at addressing these issues. These SOAR platforms integrate with multiple security tools and data sources to bring everything together and paint a clear picture of the full estate, simplifying threat detection and response actions.

For example, ABC Corp's SOAR platform works alongside its Intrusion Detection System (IDS) to generate alerts when a new strain of ransomware attempts to execute on any endpoint. This automation is just the beginning of a broader process, where technology can handle incidents far faster than manual intervention. It simply makes sense.

Components of SOAR:

- **Security Orchestration:** Integrates various internal and external tools, such as vulnerability scanners, endpoint protection products, user and entity behavior analytics, firewalls, IDS/IPS, SIEM platforms, endpoint security software, and external threat intelligence feeds. This integration enhances threat detection and context assembly.
- **Security Automation:** Uses data from orchestration by the security operations center and applies it to standardize and automate some of the more commoditized processes—for example, vulnerability scans or log analysis. SOAR platforms use AI and ML to sort threatening feeds and recommend next steps. Playbooks can be prebuilt or a customized response to threats—for example, blocking malicious emails and alerting workers.
- **Security Response:** Offers a single pane of glass to plan, orchestrate, and track post-threat detection accountability, ensuring that tribal knowledge is captured regarding which applications or systems might be impacted if they are deemed compromised. It also offers the same consolidated view for posture cleanup after an

incident, helping restore normal business activities such as case management and reporting.

CASB and SOAR: A Harmonious Approach

CASB and SOAR are complementary security solutions that together deliver complete cloud protection. CASB provides visibility and control over cloud applications and data, and SOAR provides incident response/threat remediation automation. The synergy between CASB and SOAR provides a comprehensive security solution that benefits multiple usage scenarios by working in tandem from end-to-end of an attack. This integration helps reduce the security operations workload through automatic detection and intuitive response to threats. When the CASB and SOAR functions are merged, the security posture is significantly enhanced, leveraging each solution’s individual advantages. For instance, when a CASB detects a security incident, it can trigger the SOAR platform to respond automatically to those incidents. Additionally, the SOAR platform can provide context to the CASB by integrating with threat intelligence feeds, facilitating more informed decision-making. This harmonious approach ensures a far-reaching and effective security strategy.

Identity and Access Management (IAM) Policies

Identity and access management policies are among the key areas to monitor what resources different users can access in an organization. Creating consistently enforced IAM policies across all of the platforms helps prevent unauthorized access.

Crafting Consistent IAM Policies

The new security model of unifying all elements under the IAM policies is your only source of truth. It defines who can access certain resources and what they are able to do. To have a consistent IAM policy in different environment and cloud providers, your framework needs to be standardized (while still being able) flexible on design and implementation.

To create a common set of policies, the security team at ABC Corp starts by crafting some fundamental IAM policies that need to be enforced across the board, such as utilizing MFA for all users and including least-privilege access. These policies are then customized by the cloud provider based on what functionality it supports and needs to have.

Role of Identity Federation and Single Sign-On (SSO)

Identity federation and single sign-on (SSO) are essential elements in establishing a cohesive IAM policy. They provide for frictionless user authentication and authorization between systems or applications, allowing you to manage identities easily from a single dashboard in any multicloud landscape.

- **Identity Federation:** ABC Corp has implemented identity federation to integrated on-premises AD with cloud-based IAM services. Identity federation allows employees to use their corporate credentials to access cloud resources, allowing for a consistent user experience and reduced administrative overhead.
- **Single Sign-On:** ABC Corp provides access to multiple cloud services after users sign in once without requiring them to log in to each and every service separately. This capability makes it easier for users but also ensures security because it reduces the number of passwords each user has to manage and potentially leak.

Managing Privileged Access

Privilege account management within a unified policy framework is imperative to reduce the risk of high-level access. This is achieved through the use of regulations and monitoring, which impose controls around how privileged accounts are used.

- **One Policy for All:** To manage privileged access across a multicloud environment, ABC Corp uses a unified policy framework. This framework has MFA required for all privileged accounts, access reviews performed periodically, and just-in-time (JIT) admin access provisions to minimize the duration of privileged activity.

- **Monitoring and Auditing:** Continuous monitoring of privileged account activities is critical. ABC Corp uses centralized logging and monitoring tools to track all privileged access activities in its cloud environments. This enables its PAM system (or SIEM) to quickly identify unauthorized or suspicious actions.

Implementing Multicloud IAM

Implementing a multicloud IAM strategy can be challenging due to each cloud provider's unique IAM model. However, by leveraging identity federation and centralizing identity management, organizations can achieve a cohesive IAM strategy. This approach will be detailed further in dedicated sections on federated IAM in the cloud, particularly focusing on native solutions provided by hyperscalers. To achieve unified IAM policies across diverse platforms, more agnostic approaches are required. ABC Corp utilizes an identity management solution that centralizes control and simplifies policy enforcement. This solution integrates with the IAM services of each cloud provider, enabling the security team to manage access consistently and efficiently.

By implementing this centralized identity management solution, ABC Corp can enforce its core IAM policies across all cloud environments. The solution supports federated identities and SSO, ensuring that users have seamless access to resources while maintaining strong security controls.

Regular synchronization of IAM policies across cloud platforms ensures that any changes in access controls are consistently applied. Automating the synchronization process reduces administrative burden and minimizes the risk of policy drift. Despite the complexities introduced by each cloud provider's IAM model, using federation combined with centralized identity management allows organizations to move toward an integrated IAM strategy.

Through this approach, ABC Corp ensures access is consistently managed and centrally controlled by its security team. The Itential solution enables

ABC Corporation to enforce its central IAM policies in each cloud environment, supporting federated identities and SSO to provide seamless resource access while maintaining robust security controls. Regular synchronization of IAM policies across cloud platforms ensures that access controls are always current and accurate, reflecting the latest role assignments and minimizing the risk of policy drift.

Data Protection and Privacy Policies

Aligning data governance with security policies is essential for protecting sensitive information and ensuring compliance with regulations such as GDPR, HIPAA, and California Consumer Privacy Act (CCPA). Effective data protection policies include measures like encryption and data masking, which help in safeguarding data both in transit and at rest.

Aligning Data Governance with Security Policies

A comprehensive data governance framework that aligns with security policies is crucial for any organization. This framework should ensure that all sensitive data is classified, encrypted, and subject to strict access controls. For example, a global financial institution may classify customer data, transactional records, and financial statements, ensuring each category receives the appropriate level of protection. Encryption ensures that even if data is intercepted, it remains unreadable without the decryption keys, while strict access controls limit who can view or modify sensitive information. By doing so, the company not only protects its data but also ensures compliance with various regulatory requirements.

Ensuring Compliance with Regulations

To comply with regulations like GDPR, HIPAA, and CCPA, organizations must implement robust data protection policies. These policies should include regular audits, compliance checks, and the use of encryption and data masking techniques. For instance, a healthcare provider encrypts patient records using AES-256 encryption both in transit and at rest, ensuring data remains protected against unauthorized access. Data masking techniques replace real data with fictional data during software testing to prevent exposure of sensitive information.

Network Security Policies

Network security policies are essential for protecting the integrity and confidentiality of data as it travels across networks. Segmentation policies, which divide the network into smaller, isolated segments, help contain potential breaches and prevent lateral movement of threats.

Segmentation Policies for Network Security

Many organizations employ network segmentation to create isolated segments within their networks. For example, a multinational corporation might isolate its financial systems from other parts of its network to ensure that sensitive financial data remains protected even if another part of the network is compromised. This approach not only enhances security but also improves network performance by reducing congestion.

Secure Network Configuration and Management Policies

Secure network configuration and management policies ensure that network devices and configurations are hardened against attacks. Regular audits of network configurations help identify and remediate potential vulnerabilities. For instance, a university regularly audits its network configurations to ensure all devices are running the latest firmware and software updates. Security features such as firewalls and intrusion prevention systems are properly configured to safeguard against threats.

From SDLC to SDL to SSDLC: A Journey Toward Secure Software Development

Application protection refers to the policy requirements for security and safety that are applied uniformly across the Software Development Life Cycle (SDLC). That includes documenting security consideration during development, security checks during the development lifecycle, and secure coding.

Software Development Life Cycle (SDLC): The Foundation of Application Development

The Software Development Life Cycle is either a scaffold or a planning framework by which the software is developed, deployed, and maintained. It provides a mental process to ensure that the software is delivered timely and in a cost-effective manner, and meets the functional expectations.

Origins of SDLC

The SDLC emerged in the 1960s and 1970s as the profession of software engineering began to coalesce. In the infancy of software development, early adopters sought a method that was both dependable and repeatable for handling their increasingly complex systems. The idea of an SDLC was formally introduced in a 1970 paper, “Managing the Development of Large Software Systems,” published by Winston W. Royce. This paper featured a sequential flow of development commonly referred to as the *waterfall model*.

Phases of SDLC

A traditional SDLC, like the one shown in Figure 18-1, involves developing analytics-driven software in six unique phases:

- **Planning:** Define project scope, gather requirements, and identify objectives.

- **Design:** Develop a blueprint for the system (architecture, database models, and so on).
- **Development:** Implement the code; build the software.
- **Testing:** Identify bugs, ensure quality, and validate functionality.
- **Deployment:** Release the software into production environments.
- **Maintenance:** Provide updates, fix bugs, and improve performance.

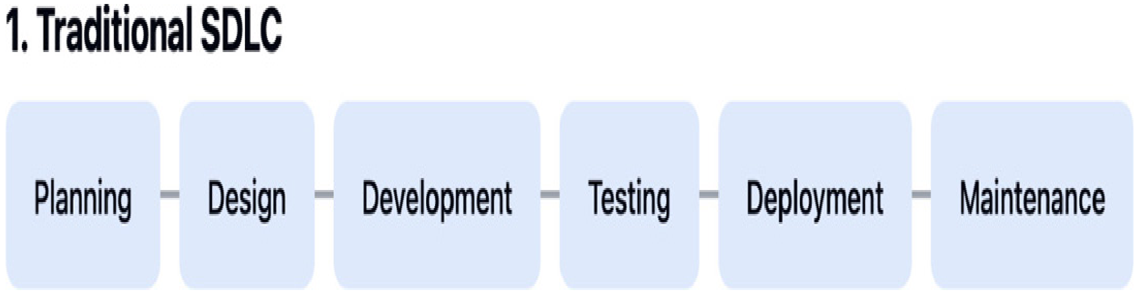


Figure 18-1 *Traditional SDLC*

Security Threat and Vulnerability Assessment and Measurement in Secure Software Development

A systematic and structured taxonomy of the most common causes of vulnerabilities is priceless. It acts both as a good anti-pattern guide for developers to watch out for and as educational material. It provides checklists that can be used in designing test cases or during audits.

The vast majority of security incidents on the Internet stem from software vulnerabilities—flaws in programs that attackers can exploit. Based on evaluations of vulnerability alerts from the Computer Emergency Response Team (CERT) and the SysAdmin, Audit, Network, and Security (SANS) Institute, as well as incident reports, it is evident that many weaknesses arise from a few common errors. Developers frequently repeat these mistakes, leading to recurring vulnerabilities. The taxonomy classifies the most common causes of vulnerabilities, helping developers, educators, and auditors enhance software security.

Common Vulnerability Causes

Vulnerabilities are critical for various use cases. For developers, a well-defined taxonomy helps identify and avoid security pitfalls. As an educational tool, it provides valuable content on security vulnerabilities for software engineering students. Finally, as a checklist for security testing and auditing, it ensures thorough and proper security testing.

- **Input Validation Failures:** Incorrect handling of input can lead to exploits such as SQL injection and cross-site scripting.
- **Authentication and Authorization Issues:** Weak or improperly implemented mechanisms can allow unauthorized access.
- **Configuration Errors:** Misconfigured systems can expose sensitive data and functionalities.
- **Error Handling:** Poor error handling can reveal system details and create entry points for attackers.
- **Cryptographic Flaws:** Using outdated or weak cryptographic algorithms can compromise data security.

Taxonomy in Action Example

Figure 18-2 and Table 18-1 depict a taxonomy of security defects and their root causes across the SDLC, aiding developers and security experts. (Please see “Security Threat and Vulnerability Assessment and Measurement in Secure Software Development” and “A Taxonomy of Causes of Software Vulnerabilities in Internet Software” as noted in the “References” sections at the end of the chapter.) When this is paired with a formal taxonomy, organizations increase the efficacy of security scanning and vulnerability remediation, resulting in more secure software applications.

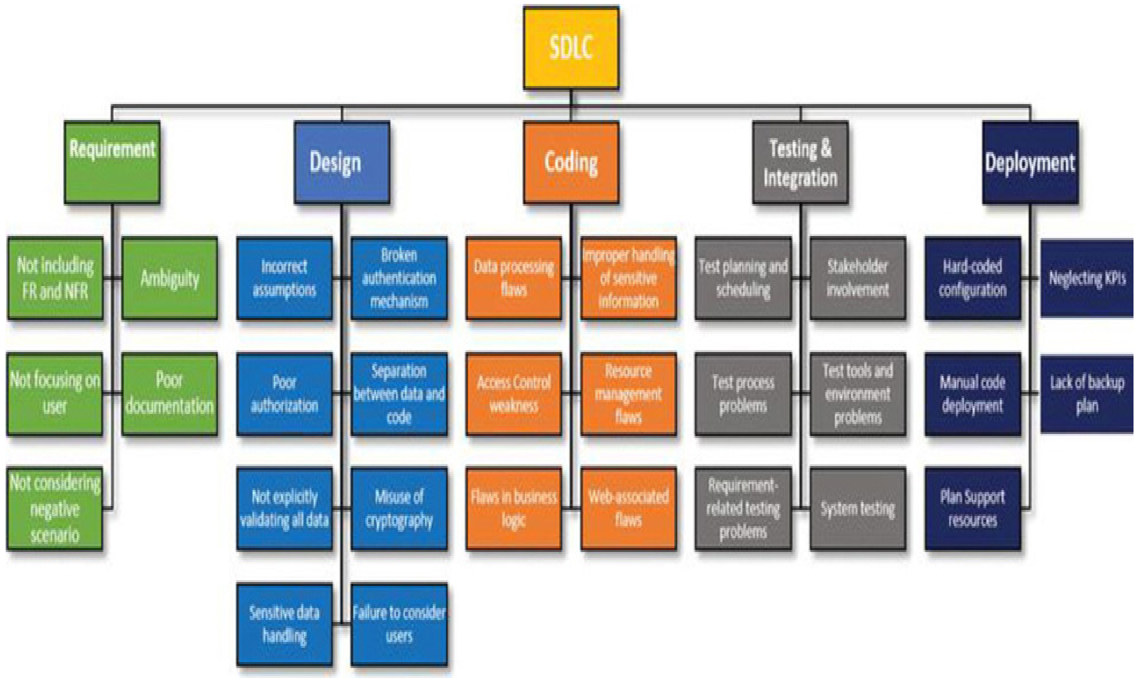
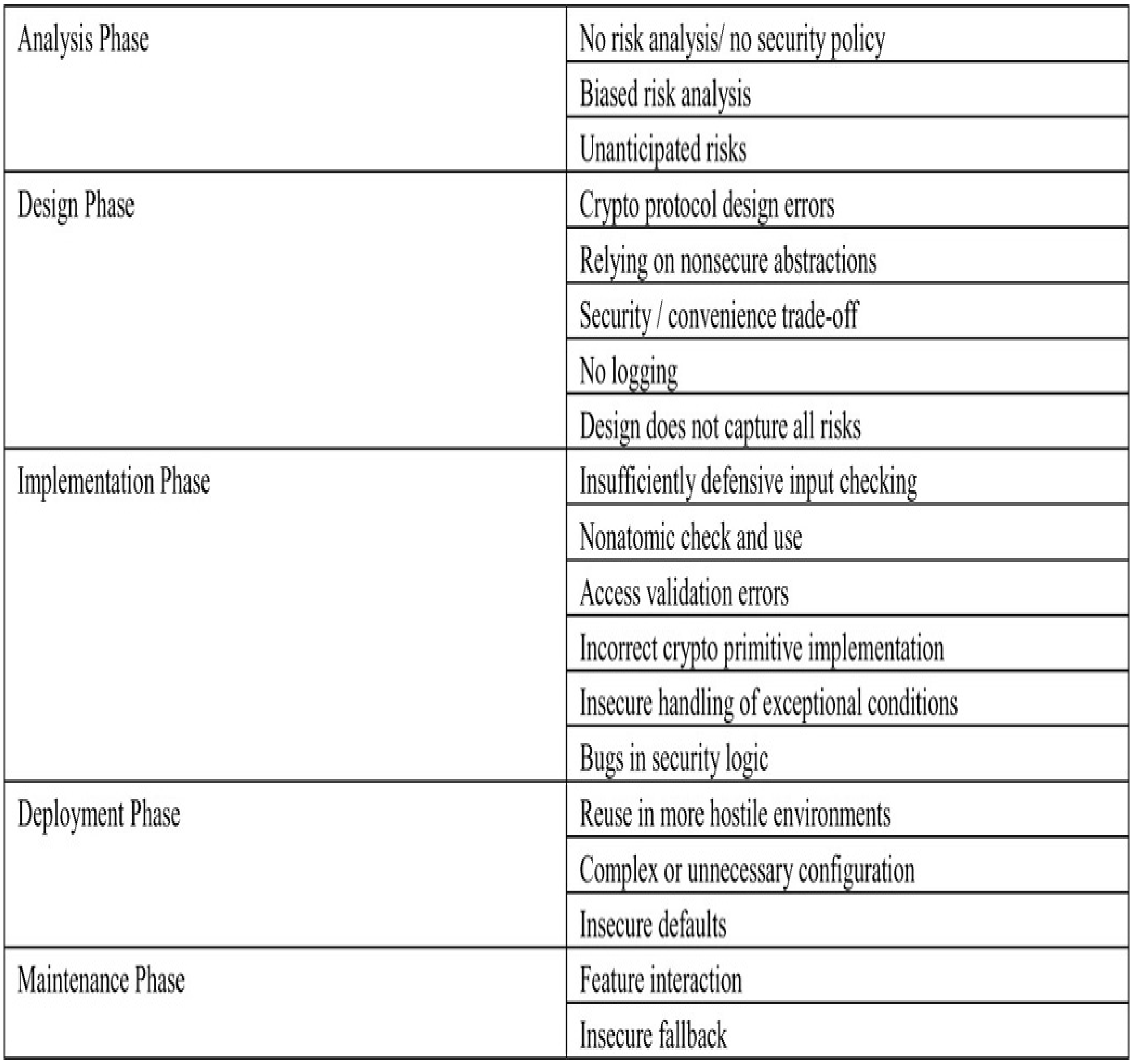


Figure 18-2 *A taxonomy of reasons for security flaws in SDLC*

Table 18-1 *A Taxonomy of Causes of Software Vulnerabilities in Internet Software*

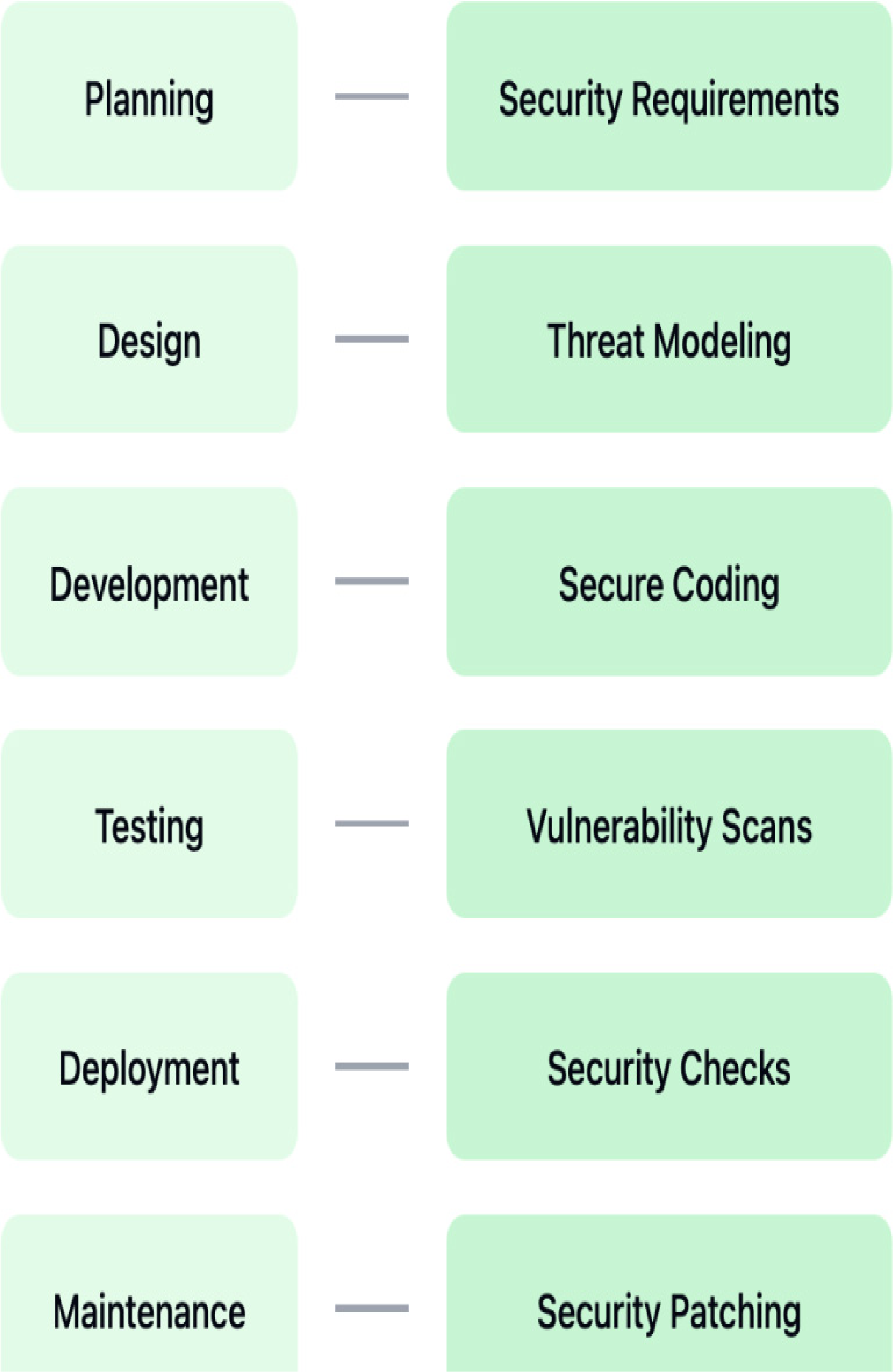


Analysis Phase	No risk analysis/ no security policy
	Biased risk analysis
	Unanticipated risks
Design Phase	Crypto protocol design errors
	Relying on nonsecure abstractions
	Security / convenience trade-off
	No logging
	Design does not capture all risks
Implementation Phase	Insufficiently defensive input checking
	Nonatomic check and use
	Access validation errors
	Incorrect crypto primitive implementation
	Insecure handling of exceptional conditions
	Bugs in security logic
Deployment Phase	Reuse in more hostile environments
	Complex or unnecessary configuration
	Insecure defaults
Maintenance Phase	Feature interaction
	Insecure fallback

Transitioning from SDLC to SDL: Embedding Security

The primary focus of the SDLC framework is on what the software does and when it will be delivered, while security is an afterthought. As a response to this gap, the Secure Development Lifecycle (SDL) was developed to incorporate proactive security practices across all phases of the SDLC (see Figure 18-3).

2. SDL (Security Enhanced)



Planning — Security Requirements

Design — Threat Modeling

Development — Secure Coding

Testing — Vulnerability Scans

Deployment — Security Checks

Maintenance — Security Patching

Figure 18-3 *Secure Development Lifecycle (SDL)*

SDL Enhancements to SDLC Phases

Following is a conceptual explanation of how SDL builds on each SDLC phase by embedding security practices:

1. **Planning Phase → Adding Security Requirements**

   The phase comprises detecting and defining the security requirements alongside the functional requirements. The objective is to define security goals and the basis of what security the system needs to enforce. Namely, this includes a set of high-level security requirements and principles, all without going into the nitty-gritty of how to implement them.

   Key actions may include activities such as identifying threats or vulnerabilities and other risks to the system, or defining compliance requirements (e.g., GDPR, PCI DSS) and encryption policies, or specifying the required authentication mechanisms (e.g., MFA) at a conceptual level.

   Example: The e-commerce platform plans to rely on secure communication (HTTPS) for user authentication and also requires protections against SQLi and XSS without getting into the technical implementation details of how these mechanisms will be implemented at this point.

2. **Design Phase → Secure Architecture and Design Reviews**

   This phase comprises implementing and validating security requirements (from the preceding Planning Phase). At this point, the security goals transform into specific architectural choices and specifications. This includes incorporating the selected mechanisms —like encryption algorithms, compliance frameworks, and authentication tactics—directly into the design of the system itself. It also includes systematic processes of the review and validation of the security design.

Following the example earlier, key actions would be choosing the specific implementations of encryption algorithms (AES-256, for example), specifying how the algorithms are to be used to protect data in transit and at rest, logging tickets for compliant architecture in line with regulations (data access controls for GDPR, and so on), and incorporating advanced authentication flows (MFA implementation, token-based auth patterns, and so on).

Example: That same e-commerce platform enforces the HTTPS requirement it derived during the Planning Phase into its architecture, by designing certificate management, enforcing TLS as a secure transport, and implementing input validation rules to thwart SQLi and XSS attacks during design.

3. **Development Phase → Secure Coding Practices**

   Singularly centered on secure coding, this phase integrates static application security testing (SAST) techniques to identify weaknesses in the codebase while being developed. Developers should also follow coding standards regarding input validation/error handling and secure communication.

   Example: For instance, using SAST, developers can avoid injection attacks by validating correctly inputted values in the course of developing the software.

4. **Testing Phase → Security Testing and Vulnerability Scanning**

   Dynamic application security testing (DAST) and penetration testing are examples of security-centric tools that can be integrated into SDL to bolster testing efforts.

   Example: An e-commerce company conducts penetration tests for its payment systems to find issues like SQL injection or XSS vulnerabilities before deploying its payment systems.

5. **Deployment Phase → Security Validation in CI/CD Pipelines**

   Part of the deployment phase integrates a final security review to verify that all security requirements were implemented. Secure release is automated by integrating configuration management tools and security checks into continuous integration and continuous deployment (CI/CD) pipelines.

Example: Before containerized applications are deployed, vulnerability scanning tools are used.

6. **Maintenance Phase → Continuous Monitoring and Incident Response**

   After the application is deployed, SDL encourages proactive patch management and continuous monitoring (with SIEM and runtime application self-protection [RASP]). Incident response plans are refined and updated to handle new threats effectively.

   Example: A healthcare organization might use SIEM to identify attempts at unauthorized access.

Why the Enhancements of SDL Are Beneficial

SDL enhances the SDLC phases by mitigating situation types and the taxonomy of common vulnerabilities that often result from traditional approaches. For example:

- **Input Validation Failures:** Mitigated in the development phase through secure coding practices and static code analysis tools.
- **Authentication and Authorization Issues:** Addressed at design time via threat modeling and secure design.
- **Configuration Errors:** Mitigated during deployment by validating secure configurations in CI/CD pipelines.
- **Cryptographic Flaws:** Avoided during the design phase with proper encryption protocols.

Secure Software Development Lifecycle (SSDLC): Evolving with Agile and DevOps

The Secure Software Development Lifecycle (SSDLC) is the evolution of the Secure Development Lifecycle (SDL) for modern workflows like Agile, DevOps, and CI/CD pipelines (see Figure 18-4). In contrast to SDL's rather linear approach, SSDLC is iterative and progressive; it stresses the need for ongoing security integration at every stage in the software lifecycle. SSDLC embeds security at every step of the application lifecycle—from development to testing to production—to help proactively find and quickly fix and prevent vulnerabilities.

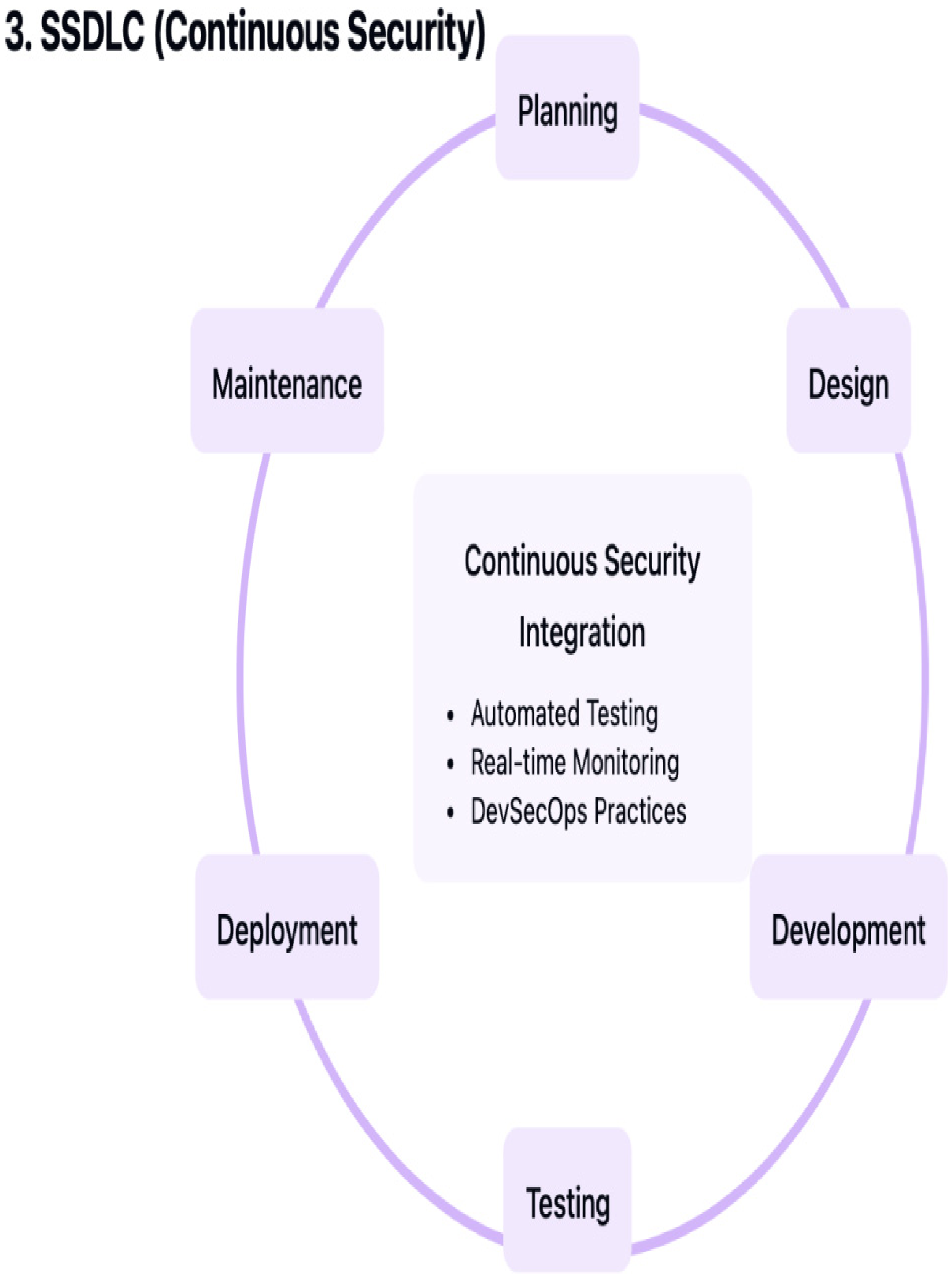


Figure 18-4 *Software Development Lifecycle Evolution: SSDLC*

Key Enhancements in SSDLC

SSDLC expands on the principles of SDL with the following:

1. **Continuous Monitoring and Feedback Loops**

   In SSDLC, security doesn't stop at deployment but extends into post-deployment monitoring. Continuous monitoring tools provide real-time insights into security events, application behavior, and potential threats. This enables teams to detect and remediate vulnerabilities dynamically.

   Example: A financial institution uses runtime application self-protection (RASP)—a security technology that protects applications in production by monitoring those applications for abnormal activity—say an attempt to access sensitive data with a privileged account.

   Feedback loops occur when security incidents identified in production environments loop back to the development and design stages. These insights inform threat models, optimize security controls, and strengthen future releases.

2. **Shift-Left Security**

   SSDLC follows the shift-left security practice in which security testing starts early in the lifecycle and happens in cycles. It minimizes the cost and effort associated with remediation by finding these vulnerabilities earlier on in the development cycle.

   Automated Tools: SSDLC integrates technologies such as static application security testing and dynamic application security testing into CI/CD pipelines. These tools help automate vulnerability scanning as part of code commits, builds, and deployments.

   Example: A software team uses GitHub Advanced Security to run SAST scans when any pull request is made, and the security scans become part of the standard code review process.

3. **Automation in CI/CD Pipelines**

   It is automation that serves as the backbone of SSDLC; it enables the security practices to match the fast-forward development cycles of Agile and DevOps teams. Security tools integrated in the DevOps toolchain ensure that every build, deployment, and production environment is security compliant.

Examples of Automation:

- **Infrastructure as Code (IaC) Security:** Terraform Cloud tests the infrastructure configuration for security and guards against policy violations before provisioning and making the resource misconfigured and less secure.
- **Container Security:** SSDLC leverages tools like Aqua Security and Cisco Cloud Application Security or Trivy to scan container images for vulnerabilities prior to deployment.

4. **Dynamic Security Testing**

SSDLC does not end with static analysis; it eliminates security test cases with a dynamic, continuous identification against production environments. This includes

- **Penetration Testing:** Frequent manual and automated penetration security testing involves real-world simulated attacks against your systems to determine your defenses.
- **Real-Time Threat Simulation:** Chaos engineering–type solutions simulate attacks in production environments and evaluate the systems on how they respond.

5. **Post-Deployment Security and Incident Response**

SSDLC maintains security as a priority after deployment with real-time monitoring, automated incident response, and threat intelligence–fueled continuous improvement.

Post-Deployment Tools and Practices:

- **Security Information and Event Management (SIEM):** Tools such as Splunk and Elastic Security provide real-time analysis of security alerts generated by hardware and applications.
- **Incident Response:** Predefined playbooks allow teams to rapidly isolate threats and remediate breaches.

Example: A retail company uses SOAR tools to automatically quarantine compromised systems and alert security teams.

Phases in SSDLC

The iterative framework of SSDLC ensures that security is a part of all stages of development and operations. Feedback loops establish a cycle of ongoing improvement:

- **Planning and Design:** Threat models are developed or modified on the fly as the requirements change. Each sprint defines security objectives according to business objectives.

  Example: An e-commerce platform defines security requirements for new features, such as enhanced encryption for user data.

- **Development:** CI/CD pipelines run automated security tools like SAST and DAST, and developers follow secure coding practices in a DevSecOps approach.

- **Testing:** Automated tools and manual penetration testing discover vulnerabilities during and post-development.

  Example: A penetration test reveals an unprotected API endpoint, which is patched before production.

- **Deployment and Monitoring:** Real-time application monitoring using RASP, SIEM, and cloud-native security tools with automated validations provide end-to-end tooling to ensure configurations and access controls are secure throughout the transition.

- **Incident Response and Feedback:** Findings from security incidents loop back into the development and design process, enhancing threat models and reinforcing subsequent versions.

  Example: A breached user account is isolated, and the vulnerability exploited is added to the next sprint’s backlog.

Benefits of SSDLC

SSDLC addresses the limitations of traditional SDLC and SDL by providing

- **Agility Without Sacrificing Security:** SSDLC allows organizations to maintain security without slowing down development cycles, making it ideal for Agile and DevOps environments.
- **Improved Risk Mitigation:** Continuous oversight and incremental testing help diminish the risk of undetected vulnerabilities.
- **Automation at Scale:** SSDLC integrates automation in CI/CD pipelines, so security practices scale as development does.
- **Enhanced Post-Deployment Security:** Continuous monitoring and automatic incident response help the organization detect and respond to threats quickly.

Understanding the Evolution of Software Development Security Frameworks

The evolution of software development frameworks reflects the growing emphasis on *security integration* as an essential component of the development lifecycle. The journey from the Software Development Life Cycle (SDLC) to the Secure Development Lifecycle (SDL) and finally to the Secure Software Development Life Cycle (SSDLC) demonstrates a gradual shift from functionality-focused processes to *continuous, automated, and adaptive security practices*.

Key Transition Milestones

The progression from SDLC to SDL to SSDLC represents a paradigm shift in software development. Each stage builds on the previous, evolving from reactive security measures to proactive integration and, finally, to *continuous, automated security practices* tailored for modern Agile and DevOps workflows. SSDLC's iterative and adaptive nature ensures that security becomes a core component of the SDLC, enabling organizations to build resilient, secure applications in an ever-evolving threat landscape.

- **From SDLC to SDL**

  1. Security Requirements Integration:

     - Define security requirements alongside functional requirements.
     - Introduce threat modeling in the design phase.
     - Establish secure coding standards.

  2. Team Collaboration:

     - Train developers on secure development practices.
     - Integrate security experts into development teams.
     - Introduce checkpoints to validate security at each phase.

- **From SDL to SSDLC**

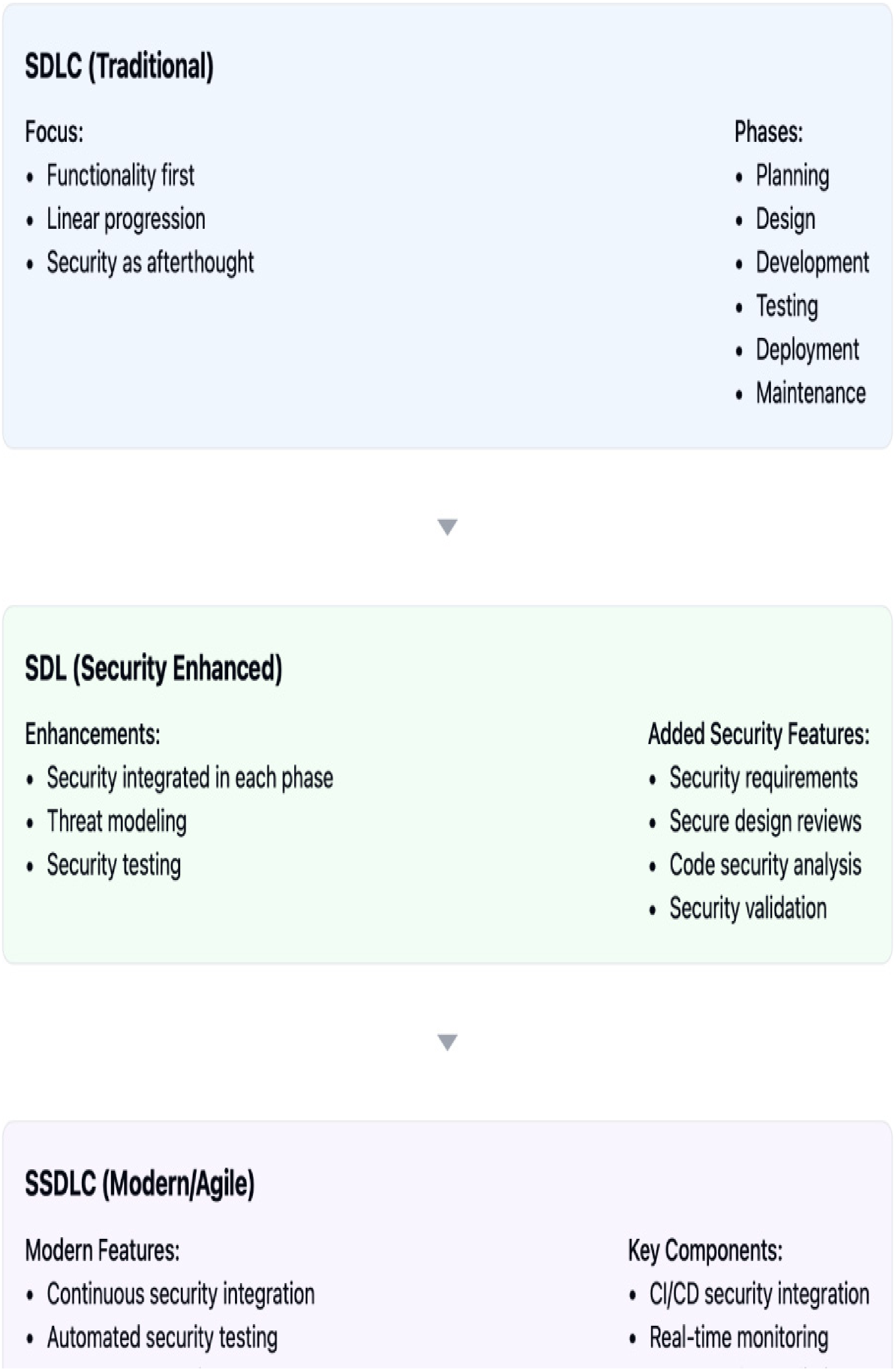
  1. Automation and Tool Integration:

     - Implement automated security testing (for example, SAST/DAST).
     - Embed security tools into CI/CD pipelines.
     - Leverage continuous monitoring tools for dynamic threat detection.

  2. Cultural Transformation:

     - Adopt DevSecOps principles to foster cross-functional collaboration.
     - Establish feedback loops between development, security, and operations teams.
     - Emphasize iterative improvement and continuous security integration.

Figure 18-5 illustrates the progression of these frameworks and their key characteristics.



SDLC (Traditional)

Focus:

- Functionality first
- Linear progression
- Security as afterthought

Phases:

- Planning
- Design
- Development
- Testing
- Deployment
- Maintenance

▼

SDL (Security Enhanced)

Enhancements:

- Security integrated in each phase
- Threat modeling
- Security testing

Added Security Features:

- Security requirements
- Secure design reviews
- Code security analysis
- Security validation

▼

SSDLC (Modern/Agile)

Modern Features:

- Continuous security integration
- Automated security testing

Key Components:

- CI/CD security integration
- Real-time monitoring

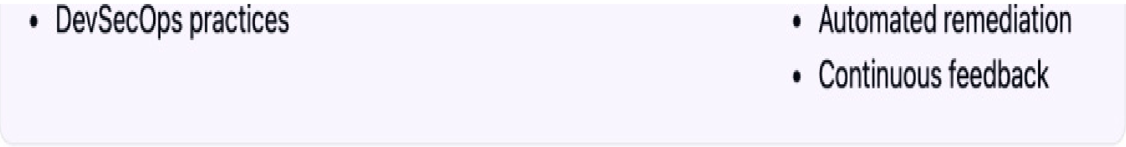
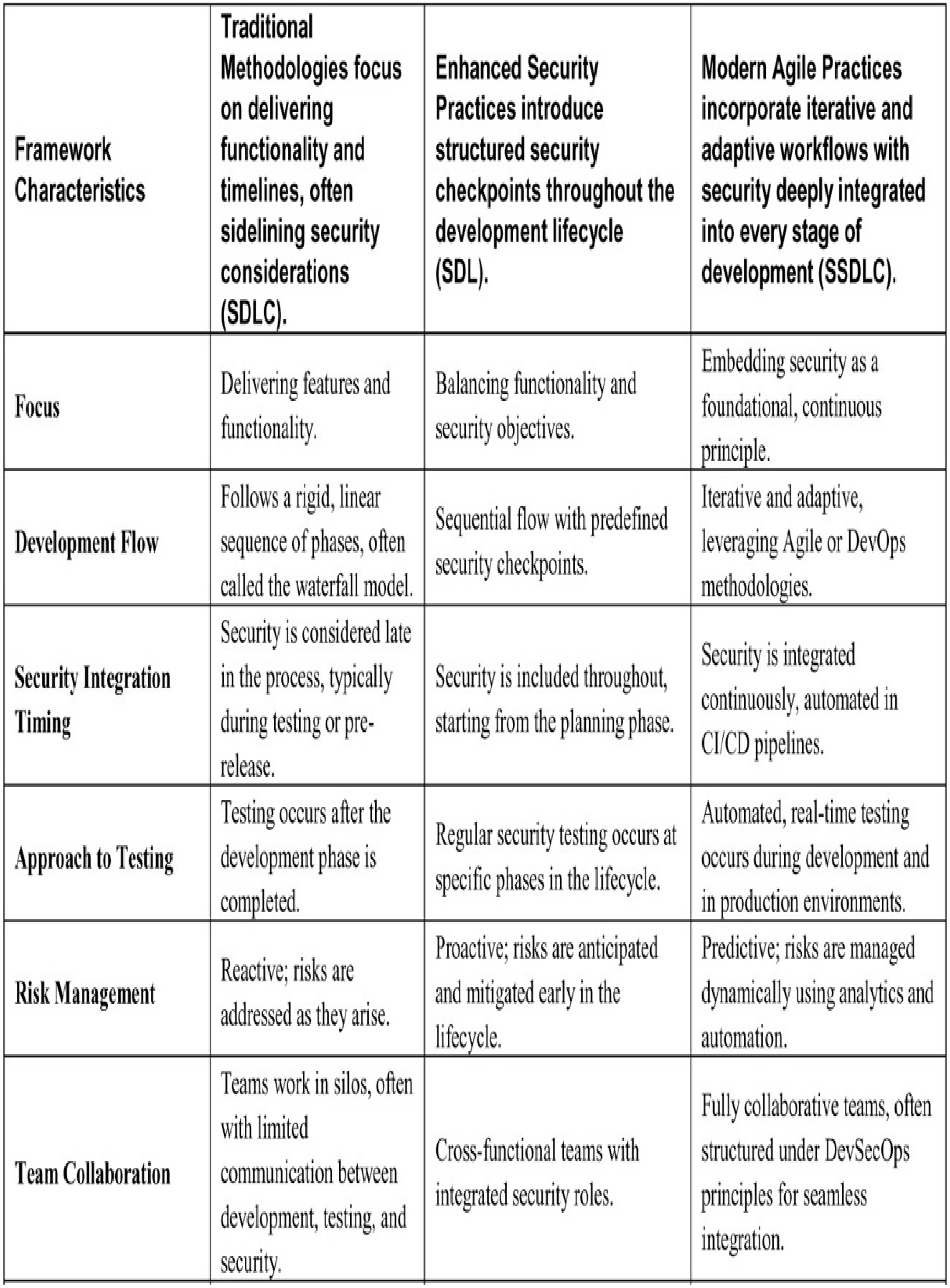


Figure 18-5 *SDLC → SDL → SSDLC*

Framework Comparison

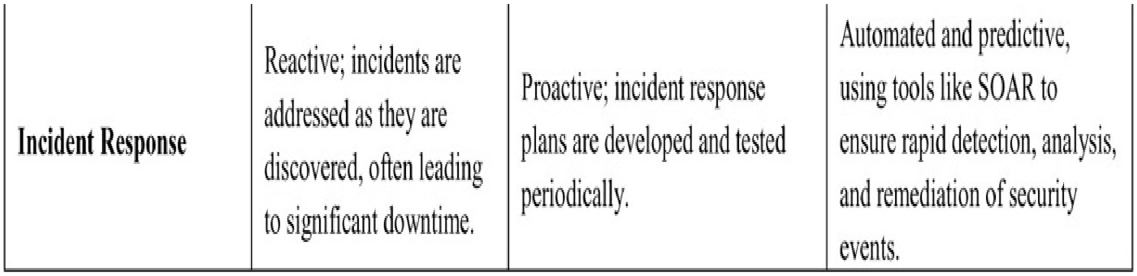
Table 18-2 provides a detailed comparison of these frameworks across various aspects:

Table 18-2 *Comparing SDLC, SDL, and SSDLC*



Framework Characteristics	Traditional Methodologies focus on delivering functionality and timelines, often sidelining security considerations (SDLC).	Enhanced Security Practices introduce structured security checkpoints throughout the development lifecycle (SDL).	Modern Agile Practices incorporate iterative and adaptive workflows with security deeply integrated into every stage of development (SSDLC).
Focus	Delivering features and functionality.	Balancing functionality and security objectives.	Embedding security as a foundational, continuous principle.
Development Flow	Follows a rigid, linear sequence of phases, often called the waterfall model.	Sequential flow with predefined security checkpoints.	Iterative and adaptive, leveraging Agile or DevOps methodologies.
Security Integration Timing	Security is considered late in the process, typically during testing or pre-release.	Security is included throughout, starting from the planning phase.	Security is integrated continuously, automated in CI/CD pipelines.
Approach to Testing	Testing occurs after the development phase is completed.	Regular security testing occurs at specific phases in the lifecycle.	Automated, real-time testing occurs during development and in production environments.
Risk Management	Reactive; risks are addressed as they arise.	Proactive; risks are anticipated and mitigated early in the lifecycle.	Predictive; risks are managed dynamically using analytics and automation.
Team Collaboration	Teams work in silos, often with limited communication between development, testing, and security.	Cross-functional teams with integrated security roles.	Fully collaborative teams, often structured under DevSecOps principles for seamless integration.

Release Cadence	Long, fixed release cycles often spanning months or even years.	Planned, moderate cycles designed to incorporate regular updates and improvements.	Short, continuous cycles with rapid iterations and incremental releases.
Adaptability	Limited flexibility; adapting to change is time-intensive and costly.	Moderate flexibility; adjustments to security or functionality require some effort.	High flexibility, with real-time adjustments to both security measures and development priorities.
Tooling	Basic development tools with minimal focus on security capabilities.	Integration of security-specific tools like vulnerability scanners, secure coding tools, and pen-testing software.	Advanced automated toolchains, including SAST, DAST, and runtime application protection tools integrated directly into CI/CD pipelines.
Cost of Fixing Vulnerabilities	High, as vulnerabilities are discovered late and require significant rework.	Medium, due to earlier detection and resolution of issues.	Low, as vulnerabilities are prevented or mitigated early in iterative cycles, reducing both time and resource costs.
Testing Automation	Rarely automated; relies heavily on manual testing efforts.	Partial automation in areas like vulnerability scanning and penetration testing.	Highly automated, with comprehensive use of tools for static and dynamic testing, integrated directly into the development and deployment pipelines.
Governance and Policy Implementation	Governance is often fragmented or nonexistent; policies are loosely defined.	Governance includes formalized policies and compliance standards.	Policies are continuously refined and enforced through automated tools, with a focus on meeting evolving compliance requirements and adapting to emerging threats.
Monitoring and Logging	Basic logging with limited or no centralized monitoring capabilities.	Centralized logging with periodic monitoring for critical systems.	Real-time monitoring integrated into SIEM tools, with automated alerts and incident response capabilities.



Incident Response	Reactive; incidents are addressed as they are discovered, often leading to significant downtime.	Proactive; incident response plans are developed and tested periodically.	Automated and predictive, using tools like SOAR to ensure rapid detection, analysis, and remediation of security events.

OWASP SAMM: A Framework for Security Maturity

As organizations transition from traditional SDLC to SDL and eventually SSDLC, ensuring robust and consistent security practices throughout the SDLC becomes paramount. The OWASP Software Assurance Maturity Model (SAMM) provides a comprehensive and framework-agnostic maturity model that helps organizations assess, improve, and optimize their software security practices.

SAMM is adaptable to any development methodology—whether it's the structured, linear approach of SDLC, the security-enhanced SDL, or the iterative, automated workflows of SSDLC—making it a critical tool for guiding secure development maturity.

For more information, refer to the "OWASP SAMM Model Overview" (see the "References" section at the end of this chapter).

Understanding SAMM's Structure

The OWASP SAMM framework, as shown in Figure 18-6, is designed to guide organizations in implementing, improving, and evaluating their software security practices.

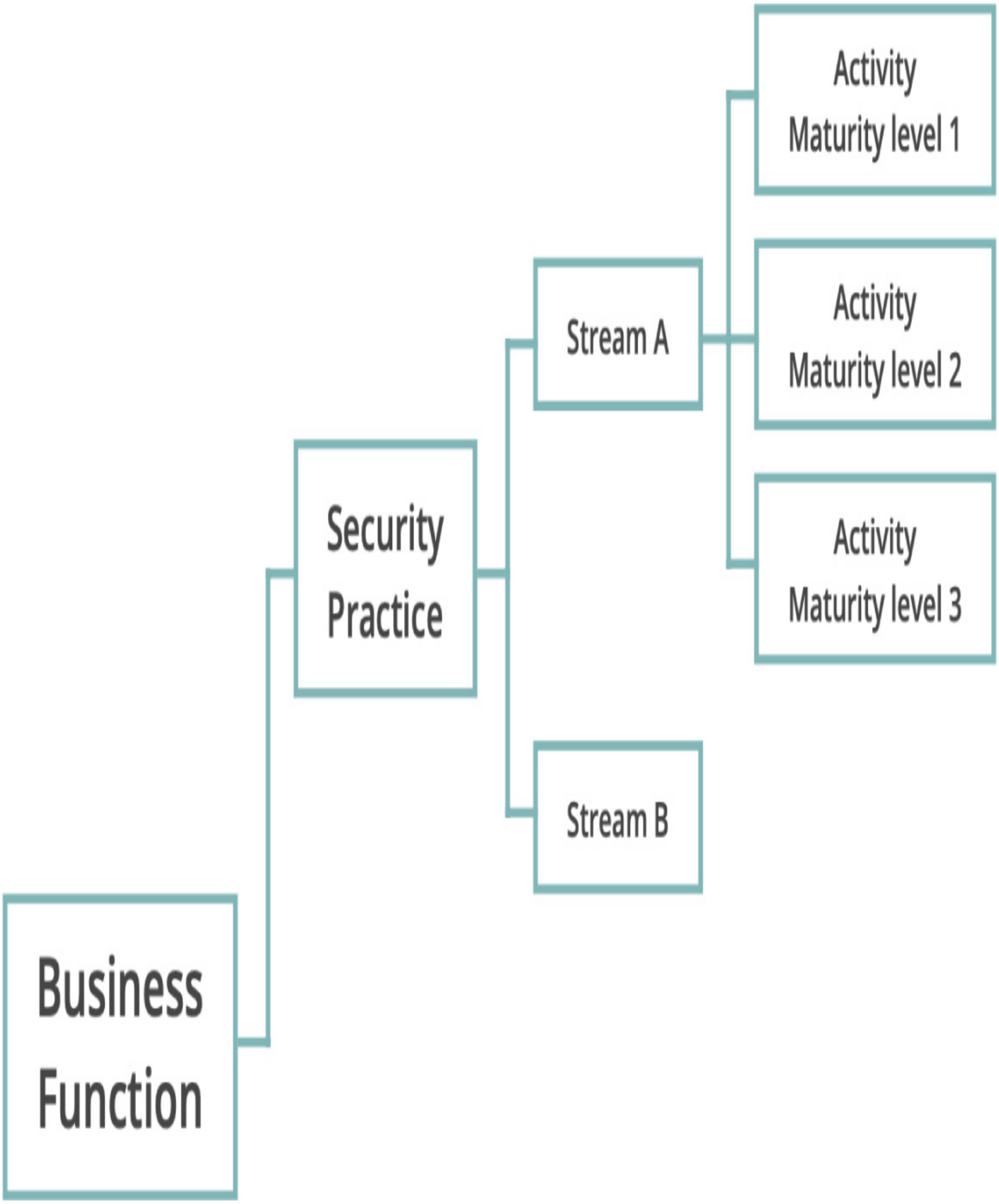

Business Function
Security Practice
Stream A
Stream B
Activity Maturity level 1
Activity Maturity level 2
Activity Maturity level 3

Figure 18-6 *OWASP SAMM Structure*

SAMM Framework Overview

The SAMM framework is structured to provide actionable guidance and measurable improvement through five core components:

1. **Business Functions:** These functions represent high-level categories of software security responsibilities.
2. **Activities:** Each business function is divided into two activities, which define specific security objectives.
3. **Streams:** Each activity contains two parallel streams, focusing on distinct yet complementary aspects of the activity.
4. **Maturity Levels:** Each stream is evaluated across three maturity levels, offering a clear roadmap for security progression.
5. **Implementation Roadmap:** Organizations can evaluate their current state, set goals, and plan improvements using SAMM's structured guidance.

The framework's strength lies in its comprehensive coverage of software security. Each business function serves a specific purpose, as seen in Figure 18-7.

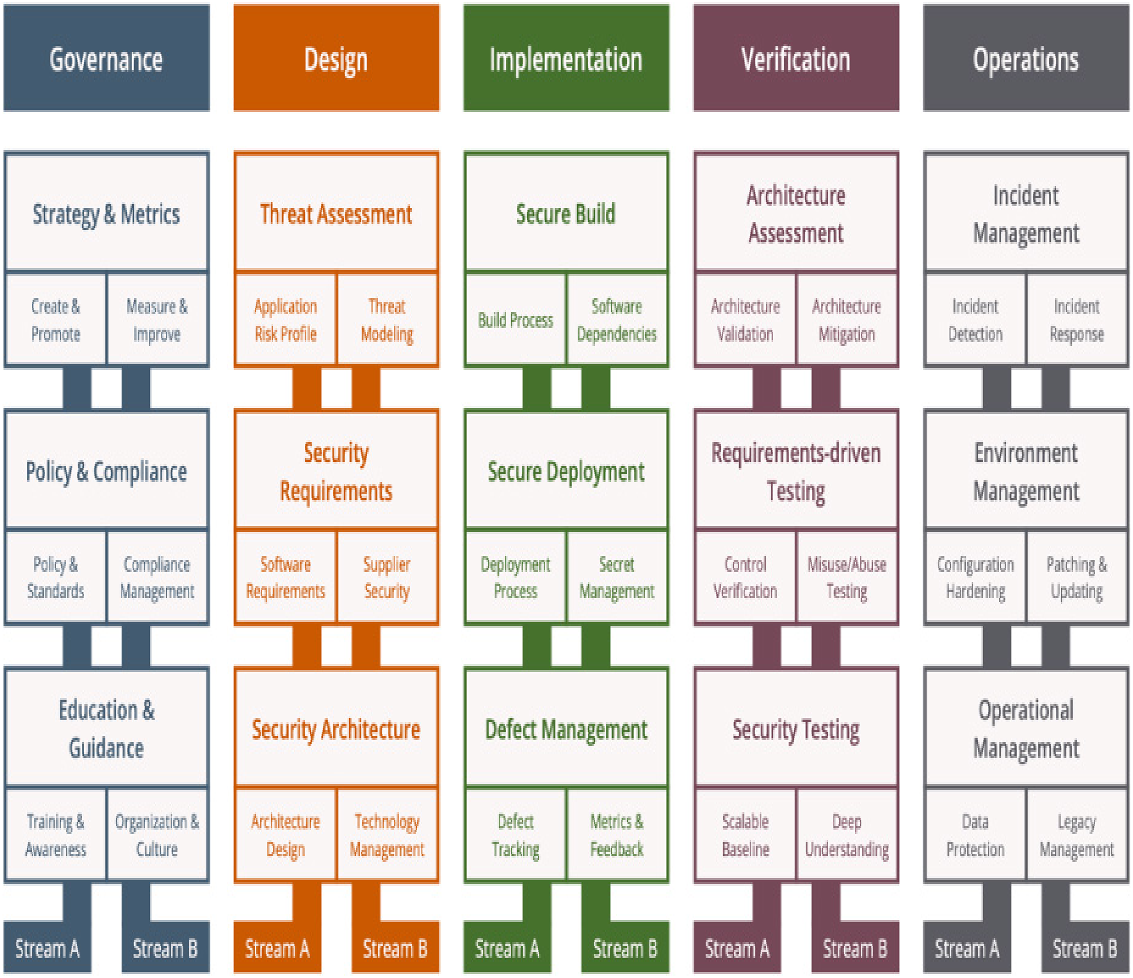


Figure 18-7 *OWASP SAMM Structure Diagram*

Business Functions

SAMM organizes secure software development into five business functions, each addressing critical aspects of software security:

1. **Governance:** Drives security strategy, metrics, policy creation, and compliance.
2. **Design:** Focuses on security requirements, threat modeling, and secure architecture.
3. **Implementation:** Ensures secure coding, secure builds, and defect management.
4. **Verification:** Validates security measures through testing and architectural reviews.

5. **Operations:** Manages security post-deployment, including monitoring and incident response.

Activities, Streams, and Maturity Levels

Each business function contains three security practices that represent different aspects of that function and can be evaluated at three levels of maturity:

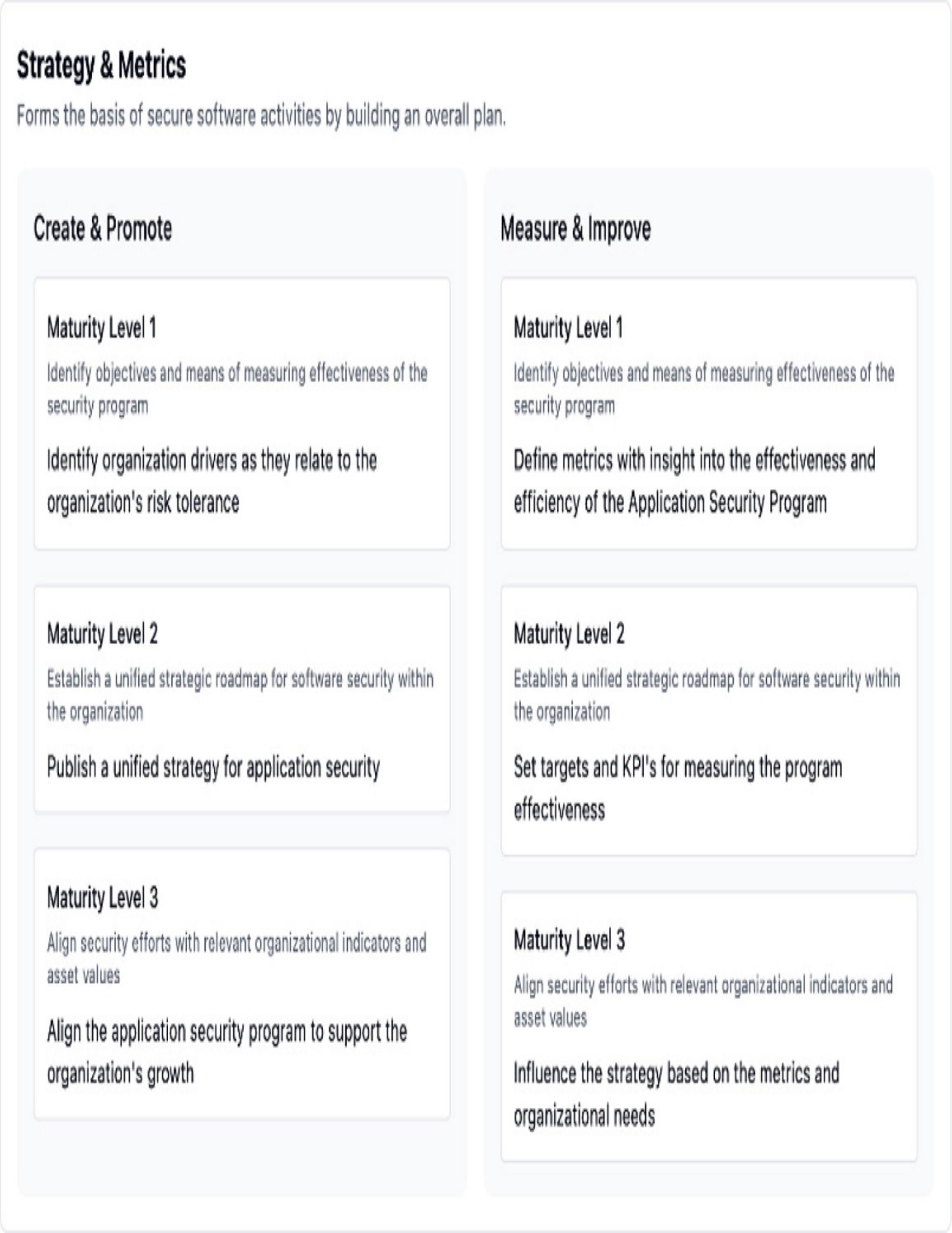
1. **Level 1 (Basic):**
   - Initial security practices implemented
   - Ad hoc processes
   - Basic understanding and implementation
2. **Level 2 (Standardized):**
   - Consistent security practices
   - Defined processes
   - Regular implementation and measurement
3. **Level 3 (Optimized):**
   - Advanced security practices
   - Automated processes
   - Continuous improvement and adaptation

Governance: A Business Function in Focus

Covering the entire SAMM framework is outside the scope of this book. However, for a better understanding of this framework, let’s analyze the Governance business function as an example of how SAMM’s structure works across activities, streams, and maturity levels. It focuses on processes and activities for managing overall software development activities, impacting cross-functional groups and organization-level business processes.

Governance ensures the organization defines and enforces a security strategy while creating policies and educating teams. Each business function contains three security practices that represent different aspects of that function. For example, the Governance function includes

1. **Strategy and Metrics:** Define security program strategy (see Figure 18-8).



Strategy & Metrics

Forms the basis of secure software activities by building an overall plan.

Create & Promote

Maturity Level 1

Identify objectives and means of measuring effectiveness of the security program

Identify organization drivers as they relate to the organization's risk tolerance

Maturity Level 2

Establish a unified strategic roadmap for software security within the organization

Publish a unified strategy for application security

Maturity Level 3

Align security efforts with relevant organizational indicators and asset values

Align the application security program to support the organization's growth

Measure & Improve

Maturity Level 1

Identify objectives and means of measuring effectiveness of the security program

Define metrics with insight into the effectiveness and efficiency of the Application Security Program

Maturity Level 2

Establish a unified strategic roadmap for software security within the organization

Set targets and KPI's for measuring the program effectiveness

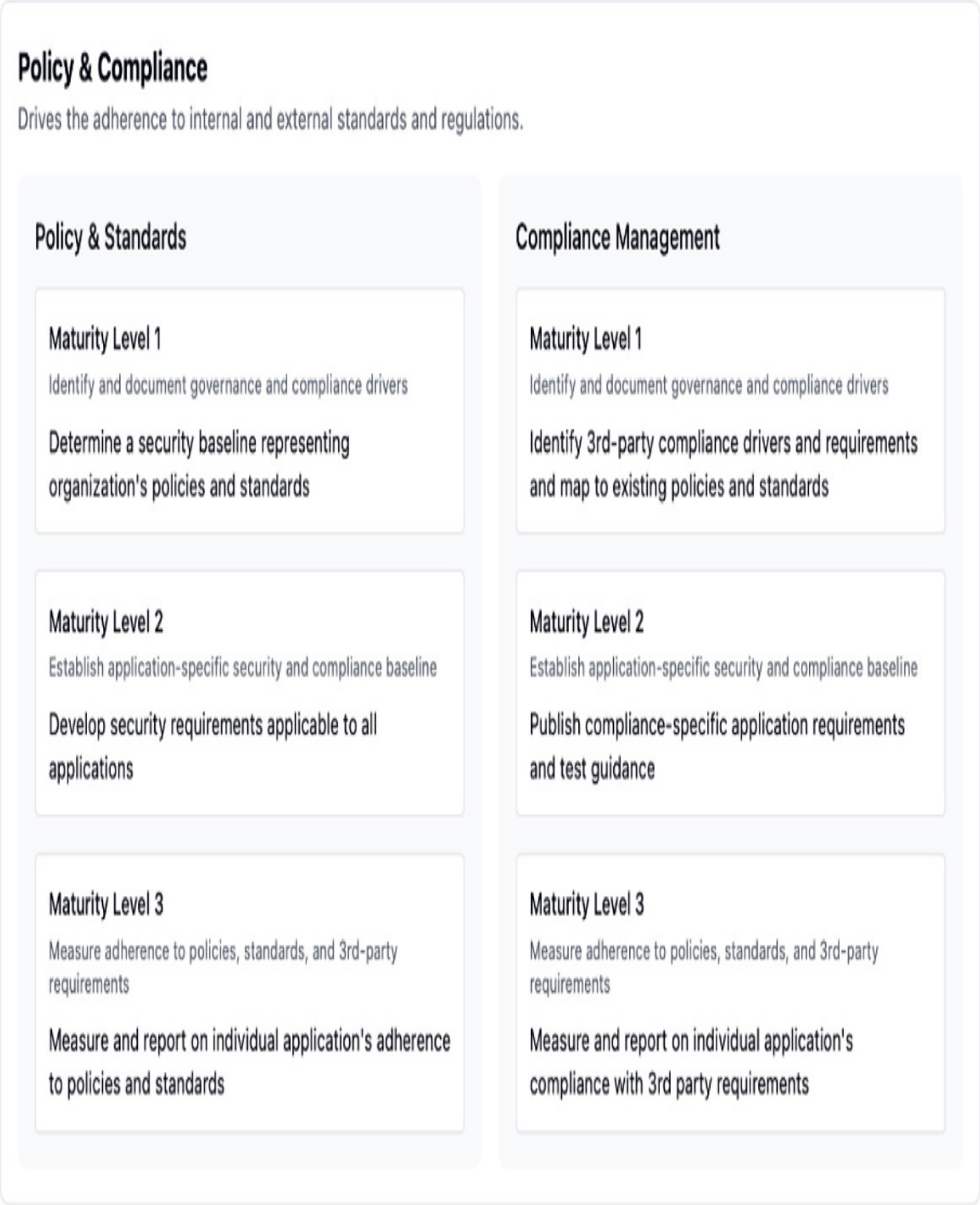
Maturity Level 3

Align security efforts with relevant organizational indicators and asset values

Influence the strategy based on the metrics and organizational needs

Figure 18-8 *Strategy and Metrics*

2. **Policy and Compliance:** Establish security policies (see Figure 18-9).



Policy & Compliance

Drives the adherence to internal and external standards and regulations.

Policy & Standards

Maturity Level 1

Identify and document governance and compliance drivers

Determine a security baseline representing organization's policies and standards

Maturity Level 2

Establish application-specific security and compliance baseline

Develop security requirements applicable to all applications

Maturity Level 3

Measure adherence to policies, standards, and 3rd-party requirements

Measure and report on individual application's adherence to policies and standards

Compliance Management

Maturity Level 1

Identify and document governance and compliance drivers

Identify 3rd-party compliance drivers and requirements and map to existing policies and standards

Maturity Level 2

Establish application-specific security and compliance baseline

Publish compliance-specific application requirements and test guidance

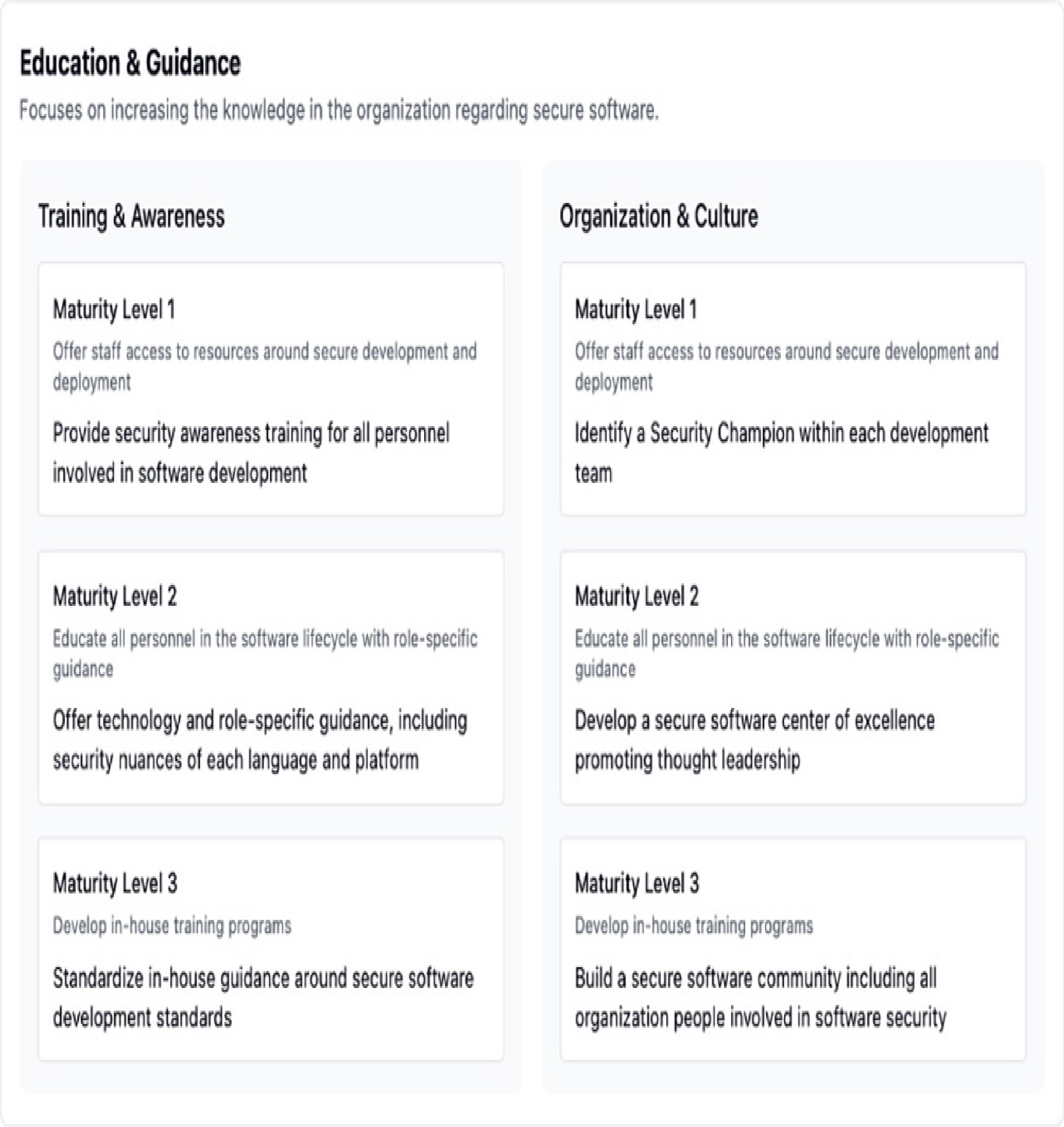
Maturity Level 3

Measure adherence to policies, standards, and 3rd-party requirements

Measure and report on individual application's compliance with 3rd party requirements

Figure 18-9 *Policy and Compliance*

3. **Education and Guidance:** Develop security training (see Figure 18-10).



Education & Guidance

Focuses on increasing the knowledge in the organization regarding secure software.

Training & Awareness

Maturity Level 1

Offer staff access to resources around secure development and deployment

Provide security awareness training for all personnel involved in software development

Maturity Level 2

Educate all personnel in the software lifecycle with role-specific guidance

Offer technology and role-specific guidance, including security nuances of each language and platform

Maturity Level 3

Develop in-house training programs

Standardize in-house guidance around secure software development standards

Organization & Culture

Maturity Level 1

Offer staff access to resources around secure development and deployment

Identify a Security Champion within each development team

Maturity Level 2

Educate all personnel in the software lifecycle with role-specific guidance

Develop a secure software center of excellence promoting thought leadership

Maturity Level 3

Develop in-house training programs

Build a secure software community including all organization people involved in software security

Figure 18-10 *Education and Guidance*

How SAMM Relates to SDLC, SDL, and SSDLC

The OWASP SAMM framework, as shown in Figure 18-11, works with any application development methodology, each of which has its unique challenges and benefits. It bridges the security gaps inherent in traditional software development approaches, offering measurable progress as organizations mature from SDLC to SDL to SSDLC.

Traditional SDLC considers security usually at the end of the SDLC, whereas SAMM offers the solutions by necessitating the development of security practices in earlier stages like creating policies, threat modeling, and secure architectural design because risk mitigation is always better to be performed beforehand. While this approach is structured through various stages of the lifecycle, SAMM, with its maturity metrics, augments governance, secure coding, and vulnerability testing practices in SDL.

SAMM fits seamlessly into iterative and automated workflows, enabling both automation and real-time monitoring and incident response in CI/CD pipelines, making it a good fit for modern SSDLC methodologies like Agile and DevOps. SAMM allows organizations to implement security in a holistic and ongoing fashion throughout their development lifecycle by accommodating these various frameworks.

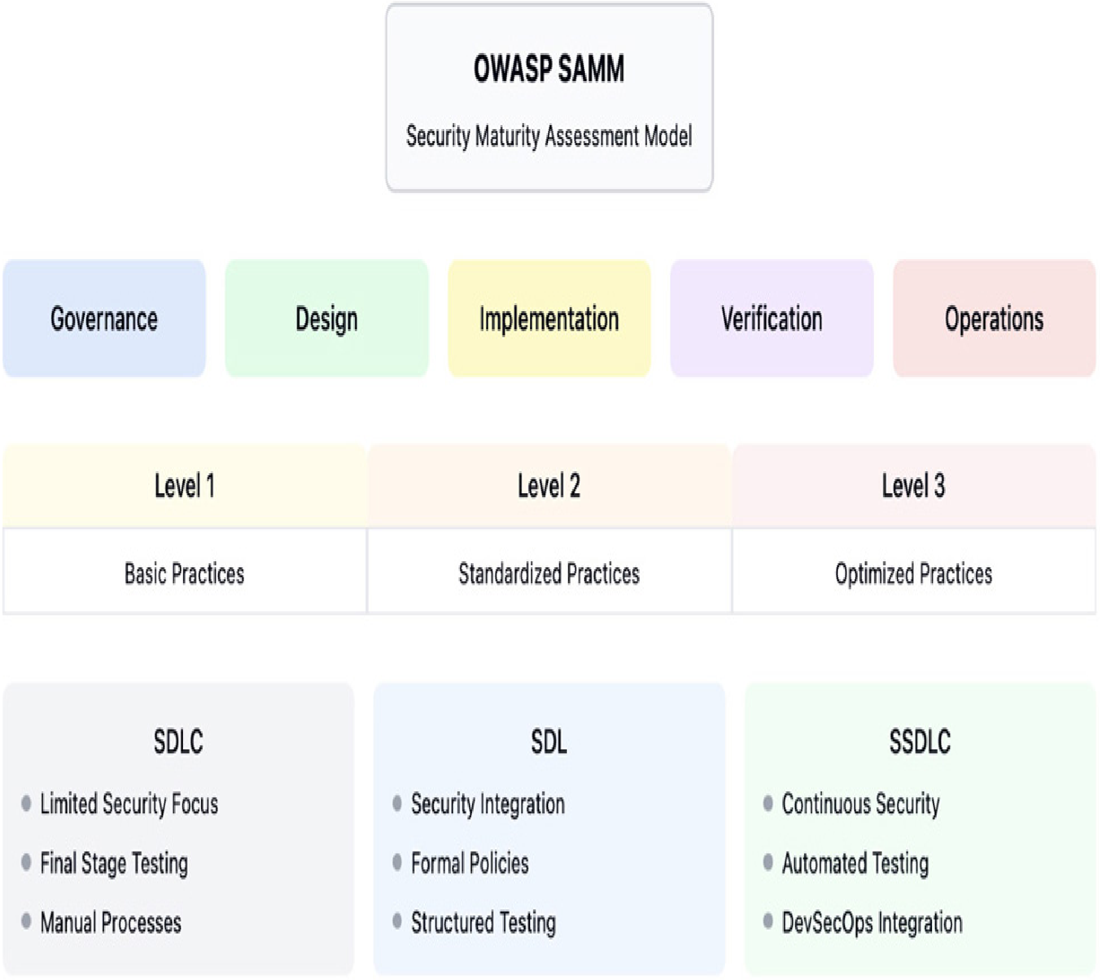
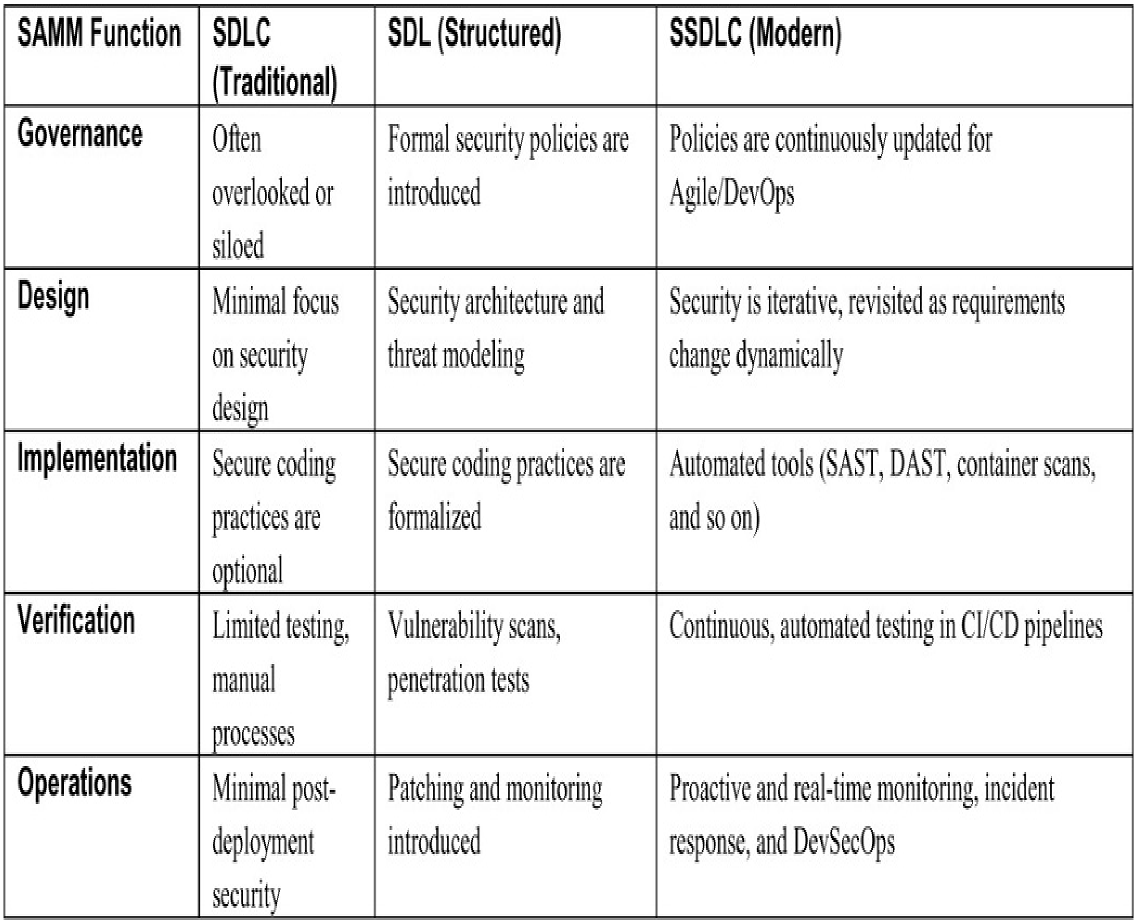


Figure 18-11 *OWASP SAMM Framework and Relationships*

Table 18-3 shows how SAMM fits across frameworks.

Table 18-3 *How SAMM Fits Across Frameworks*



SAMM Function	SDLC (Traditional)	SDL (Structured)	SSDLC (Modern)
Governance	Often overlooked or siloed	Formal security policies are introduced	Policies are continuously updated for Agile/DevOps
Design	Minimal focus on security design	Security architecture and threat modeling	Security is iterative, revisited as requirements change dynamically
Implementation	Secure coding practices are optional	Secure coding practices are formalized	Automated tools (SAST, DAST, container scans, and so on)
Verification	Limited testing, manual processes	Vulnerability scans, penetration tests	Continuous, automated testing in CI/CD pipelines
Operations	Minimal post-deployment security	Patching and monitoring introduced	Proactive and real-time monitoring, incident response, and DevSecOps

Benefits and Implementation

To sum it up, the OWASP SAMM framework provides a structured way for organizations to incrementally improve their software security posture by leveraging these features of measurable progress, flexibility, and coverage. Some of its primary benefits are that it provides a transparent view of maturity levels, clear pathways for improvement, and measurable outcomes to track progress. SAMM is flexible enough to be used with any development methodology, whether it is a traditional SDLC, structured SDL, or modern SSDLC, and customized to the specific needs of an organization. SAMM is comprehensive, covering a full spectrum of governance, design, implementation, verification, and operations and is designed so that security practices are scalable and applicable across a range of organization sizes and industry sectors. In addition, SAMM supports both modern practices such as shift-left security and works seamlessly within Agile and DevOps workflows.

SAMM is implemented over three distinct phases: Assessment, where organizations assess current maturity and identify gaps; Planning, where organizations define target maturity levels, create improvement roadmaps, and allocate resources; and Execution, where changes are implemented, tracked, and tailored. By taking this deductive methodology, organizations can examine the vital spheres of their security towers while iteratively refining their craft. Having SAMM means securing each phase of the development lifecycle and aligning it with the current methodologies to keep organizations focused on security, thus making them evolve along with the issues today.

Monitoring, Logging, and Auditing Policies

Implementing unified logging standards across environments is crucial for effective monitoring and incident response. Continuous monitoring and real-time analysis policies enable organizations to detect and respond to threats promptly.

Unified Logging and Continuous Monitoring

Unified logging standards that apply across on-premises and cloud environments ensure consistent security event logging. Continuous monitoring tools analyze these logs in real time, allowing prompt detection and response to potential threats. For instance, a telecommunications company uses these tools to correlate logs from various sources, gaining a comprehensive view of its security landscape and identifying patterns that might indicate a coordinated attack.

Automated Auditing and Compliance Reporting

Automated auditing and compliance reporting tools maintain an ongoing assessment of the security posture and ensure adherence to regulatory requirements. Regular audits and compliance reports generated by these tools help identify areas for improvement. For example, a financial services firm uses these tools to audit its security policies regularly and generate compliance reports, ensuring that the firm remains compliant with relevant regulations.

Incident Response and Remediation Policies

Creating incident response frameworks that span both cloud and on-premises environments ensures that organizations can effectively respond to security incidents regardless of where they occur.

Incident Response Frameworks

Developing a comprehensive incident response framework is crucial for minimizing the impact of security incidents. This framework should include predefined procedures for detecting, responding to, and recovering from security incidents. For instance, a retail chain’s incident response plan allows it to quickly isolate affected systems, mitigate threats, and begin the recovery process, ensuring business continuity even during a breach.

Automated Policy-Based Remediation

Automated policy-based remediation tools can quickly address detected issues, reducing the impact of security breaches. If a vulnerability is detected in a critical system, remediation tools can automatically apply patches or reconfigure the system to mitigate the threat. These tools accelerate response times and reduce the likelihood of human error during the remediation process.

Policy Compliance and Verification

Continuous compliance monitoring tools are essential for ensuring that security policies are adhered to at all times. Integrating policy checks into CI/CD pipelines ensures that security is built into the development process, catching issues early before they reach production.

Continuous Compliance Monitoring

Using continuous compliance monitoring tools ensures consistent enforcement of security policies. These tools provide real-time visibility into the compliance status, allowing prompt identification and remediation of deviations from established policies. Continuous monitoring also helps detect potential security threats, enabling proactive measures.

Integrating Policy Checks into CI/CD Pipelines

Integrating policy checks into CI/CD pipelines makes security an integral part of the development process. This ensures that security issues are caught early, reducing the risk of vulnerabilities in deployed applications. Automated policy checks ensure that code is continuously tested against security standards, maintaining a secure development environment.

Challenges in Policy Enforcement Across Hybrid Environments

Many challenges are created by the enforcement of security policies across hybrid environments; they include different capabilities of cloud platforms and organizational resistance to unified frameworks.

Inconsistent Enforcement Capabilities

The security features and capabilities in different environments are often mutually exclusive, which hampers the enforcement of a common policy across diverse conditions. This gap can be filled using policy management tools that centralize the security measures and deliver a common approach to enforcing policies across your enterprise. This helps all the environments, and that is one example how ABC Corp employs them to provide consistent enforcement capabilities via those tools leveraging them to enforce a single security strategy instead of targeted on individual ones.

The Need to Overcome Resistance to Unified Frameworks

Organizational stakeholders resist unified security frameworks. It is essential to show the way a consolidated structure can bring improvement in security and operational efficiency. Training and educating about the value of a consolidated security approach, including examples that are succeeding in practice, can address this resistance and ease the process for consistent security policies across all environments.

Future Directions in Policy-Based Security

The future of policy-based security lies in predictive policy enforcement and adaptive policies that can adjust to changing conditions and emerging threats.

Predictive and Adaptive Policies

Advanced predictive and adaptive policies leverage analytics and machine learning to anticipate potential security threats and adjust policies accordingly. Artificial intelligence (AI) security solutions analyze patterns to predict where the next wave of threats may emerge, automatically adapting an organization’s security policies. These solutions are highly sought after, particularly for their ability to identify vulnerabilities before they can be exploited.

The Role of AI in Dynamic Policy Management

AI is playing a major role in dynamic policy management: AI can rapidly detect patterns in data streams in real time and identify abnormalities suggestive of potential threats. These tools provide complementary policy enforcement mechanisms that keep these security policies effective against the particular threats of today. AI is helping organizations to enhance and fine-tune security strategies in a constant battle against new threats that appear as an unending changing threat landscape.

Security Suites Delivered by the Cisco Security Cloud

Figure 18-12 unveils Cisco’s Security Cloud, which offers a suite of security solutions that provide more than just product bundles. These suites represent a cohesive integration of various business units working together to deliver robust security through enhanced integration and AI-driven capabilities. The result is a comprehensive security platform that provides effective, scalable protection for your business, users, and cloud environments.

Designed to secure every aspect of your network, Cisco Security Cloud solutions are cloud-native and cloud-delivered, powered by AI to ensure effective and scalable protection. Cisco offers three primary solutions to simplify security management: Cisco Breach Protection, Cloud Protection, and User Protection. These solutions provide a platform-driven approach that simplifies security architectures, reducing the complexity of managing multiple products. This results in better security efficacy, improved user and administrator experiences, and enhanced ROI. Whether you’re at the beginning of your security journey or looking to grow securely, these solutions can support your needs.

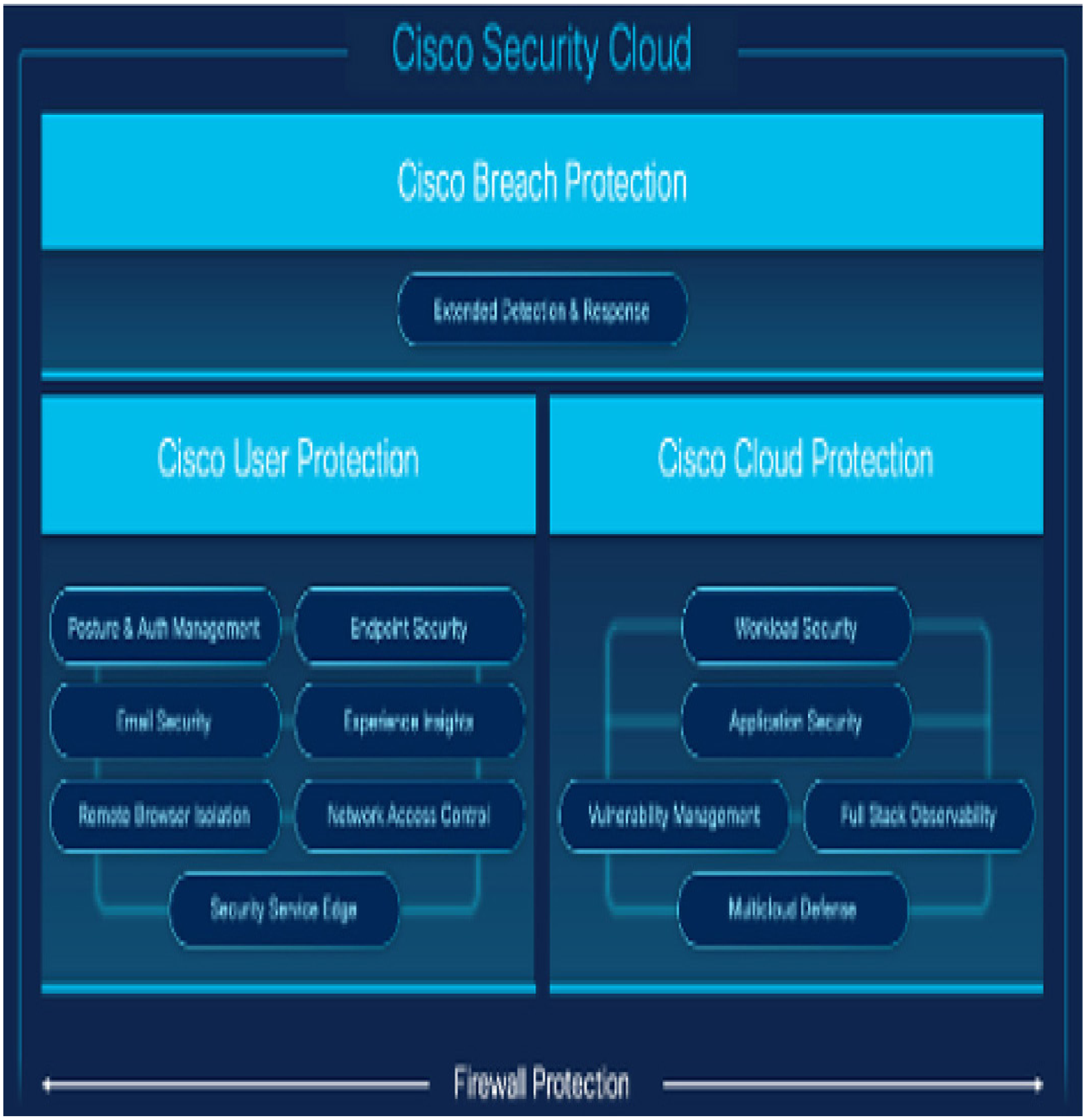


Figure 18-12 *Security Suites Delivered by the Cisco Security Cloud*

Cisco User Protection Suite

Cisco User Protection Suite ensures secure access for every user to any application, on any device, from anywhere. It implements a zero trust approach with minimal friction, allowing users to work securely while maintaining operational efficiency through better visibility and control. Cisco User Protection Suite does the following:

- **Streamlines User Security:** Instead of relying on disparate point solutions, Cisco combines tools for comprehensive and cost-effective security. This reduces coverage gaps across the environment, including applications and contract workers, thereby minimizing risk across the organization.

- **Improves Security Outcomes:** Cisco User Protection delivers accurate, rapid, and actionable threat intelligence, backed by Cisco Talos, one of the world’s largest commercial threat intelligence teams. This suite provides end-to-end user security that adapts to evolving threats, enforces least-privilege access, and proactively prevents breaches.
- **Simplifies and Scales:** Cisco User Protection delivers secure access to all applications—traditional, on-premises, and cloud—with comprehensive end-user protection that scales with your organization. It ensures security resilience across dynamic environments with a solution that adapts access policies to changing risks.

Cisco Cloud Protection Suite

Cisco Cloud Protection Suite offers end-to-end security for applications, workloads, networks, and clouds with complete coverage across on-premises, public, private, and hybrid cloud environments. In essence, the suite fortifies hybrid and multicloud security, lowers risk, and promotes compliance, regardless of your workload’s space between operating environments, in a centralized and holistic way. Cisco Cloud Protection Suite does the following:

- **Reduces Your Attack Surface:** The suite contains comprehensive application security that protects workloads running on-premises and in the cloud with granular least-privilege access. By deploying the network-based microsegmentation system, Cisco Cloud Protection Suite essentially prevents lateral movement and properly contains threats as traffic crosses clouds or moves between VPCs across networks.
- **Focuses on Risk:** By ranking threats and exposing high severity risks to the business, Cisco Cloud Protection gives you unmatched

visibility from your network to your applications, to the cloud—allowing for better decision-making.

- **Scales on Your Own Terms:** The suite enables you to reclaim ownership of your application environment with comprehensive, flexible security that scales along with you at any phase in your journey to the cloud.
- **Optimizes Multicloud Operations:** Cisco Cloud Protection helps you realize cloud economics at scale while ensuring security. You can have your cake and eat it too.

Cisco Breach Protection Suite

Cisco Breach Protection Suite streamlines operations to speed responses against breaches, combining AI and automation with a set of connected tools to confidently confront the most advanced threats.

Built on zero trust principles, Cisco Breach Protection offers results-driven security and improves speed of detection and response to advanced threats with user-first protection. This provides single-pane-of-glass and one-click access across email, endpoints, network, and cloud environments for threat detection efficacy as well as standardization in the security operations process. Cisco Breach Protection Suite does the following:

- **Increases Efficiency:** The suite enables you to use AI to prevent attacks and support machine-scale analytics for human-scale insight. SOAR enables SecOps teams to apply analytics across-domain telemetry, reducing noise in their environments and advancing beyond traditional SIEM- or endpoint detection and response (EDR)–centric approaches.
- **Delivers Ultimate SOC Experience:** More advanced contextual detection with human judgment will allow security operations analysts to make quicker and more informed response decisions.
- **Improves ROI:** Cisco Breach Protection accelerates response times to prioritized incidents, focusing on threats that pose the greatest material risk to the business. Optimized for multivendor

environments, SOC teams can remain agile and adaptive while advancing their organizational maturity.

The combination of Cisco Security Cloud solutions works together to offer extensive protective functionality, efficiency improvements in security operations, and increased information about the organization's resilience. These suites are engineered to protect businesses wherever they go, automatically providing consistent security across users, clouds, and networks.

Summary

In this chapter, we explored contemporary techniques and approaches for building secure software solutions that evolve with an ever-changing and dynamic cloud delivery model. We examined the shift from a traditional **DevSecOps continuum (via SDLC)** to a more agile **Secure Software Development Lifecycle (SSDLC)**, reinforcing stronger security policies, cloud security tools such as **CASBs and SOAR**, and **OWASP SAMM compliance** to secure hybrid and multi-cloud infrastructure. Notably, **scalability and agility** are achievable only through **automation, iterative processes, and real-time monitoring**.

Aligned with zero trust principles, this chapter emphasized continuous verification (IAM, RBAC, MFA), least-privilege access, micro-segmentation, and adaptive, AI-driven policies. Consistent governance through unified security policies, along with robust encryption and compliance frameworks, enables organizations to mitigate cloud-level data breaches. SSO and CASBs are key tools in securing access by enabling zero trust network access, fostering resilience, scalability, and agility. This reinforces zero trust principles across the software supply chain and cloud security, bringing organizations closer to a holistic zero trust architecture.

References

1. "Security Threat and Vulnerability Assessment and Measurement in Secure Software Development": https://www.researchgate.net/publication/357905624_Security_Thre

at_and_Vulnerability_Assessment_and_Measurement_in_Secure_Software_Development

2. “A Taxonomy of Causes of Software Vulnerabilities in Internet Software”: https://www.semanticscholar.org/paper/A-taxonomy-of-causes-of-software-vulnerabilities-in-Piessens/5ec693950d1e6039e04a7b86a488e816ddcdd82e

3. OWASP, “SAMM Model Overview”: https://owaspsamm.org/model/

Chapter 19. Workload Mobility: On-Prem to Cloud

In this chapter, you will learn about the following:

- Workload mobility fundamentals
- Benefits of cloud migration
- Challenges in migration
- Zero trust integration
- Migration strategies
- Hybrid and multicloud models
- Data security during migration
- Post-migration optimization
- Best practices and lessons

Definition and Scope of Workload Mobility

The point of workload mobility is to provide the capability to transport applications, services, and its data from one environment to another environment, especially from on-premises data centers to the cloud. This notion has grown from the computing burdens on servers to the full range of applications, processes, capabilities, and even business strategy. In the cloud context, a *workload* is any application, service, or measurable amount of work utilizing cloud resources like compute power, networking, and storage. The transfer is much more complicated than simple data migration; it is a holistic process that ensures the ongoing functionality of the application as well as data consistency and the actual running function of the service in a cloud environment. In the context of zero trust, workload migration plays a critical role by reinforcing the principle of “never trust, always verify.” As workloads are moved and restructured in cloud environments, maintaining robust security controls, visibility, and segmentation becomes essential to ensuring that trust boundaries are upheld at every stage of migration.

Is Your Cloud Ready for Your Workloads? Understanding the Benefits and Challenges

There are clear benefits to transitioning workloads from on-premises data centers to the cloud, but the process of migrating is complex. A detailed understanding of the challenges is critical to building a well-informed and resilient migration strategy.

Perhaps the most obvious benefit of cloud migration is the **scalability** and **flexibility** it provides. Cloud environments are highly adaptable and responsive to business needs, helping to improve performance. This benefit is particularly valuable to companies with fluctuating workloads and seasonal demands. For example, an **e-commerce company** may experience high traffic during holiday seasons. The cloud enables the company to **scale resources up** during high-traffic periods and **scale down** during off-peak times, optimizing both cost and performance.

Moving from CapEx to OpEx changes the financial model of IT spending. Transitioning from owning and maintaining hardware to paying for cloud services on a usage basis can result in significant cost savings. For example, a **startup** may need minimal computational power initially and can then **incrementally scale up** as needed, without incurring heavy upfront infrastructure costs.

However, to realize these savings, organizations must **carefully monitor resource usage** and ensure efficient allocation. This involves:

- **Avoiding over-provisioning** by accurately matching resource allocation to workload requirements,

- **Identifying and releasing stale or unused resources** across the organization's cloud tenants, and
- Implementing continuous cost optimization practices as part of a cloud FinOps strategy.

Unplanned spending and hidden costs can lead to budget surprises, making **cloud cost management** one of the top concerns for organizations adopting cloud services.

Additionally, cloud environments offer always-on, cloud-scaled, reliable disaster recovery (DR) solutions. As data and applications are replicated over numerous geographic locations, the impact of failures and disruptions at a local level is reduced. Cloud providers structure their infrastructure into *regions* (regions are mostly independent of each other) and segments of regions called *availability zones (AZs)* (each AZ is basically a set of isolated data centers within the region). This architecture allows one AZ within a region to take over resources located in other AZs in the event of a failure of a particular availability zone. As an example, every financial services company keeps near real-time copies of its business-critical databases in different availability zones within a region (or even across regions). If one data center goes offline due to a natural disaster, the firm can quickly switch to another location, ensuring continuous operations and data integrity.

While these benefits are great, moving to the cloud has its own challenges. Indeed, ensuring data security and compliance remains a challenge, especially for sensitive workloads. Businesses will require enforcement of strong security policies and comply with industry regulations during and after the cloud migration. For example, if a healthcare provider processes patient records, it must comply with Health Insurance Portability and Accountability Act (HIPAA) regulations and maintain encryption and limited access during migration.

If we look at Flexera's 2024 State of the Cloud Report, a detailed view of the top challenges for cloud migration is revealed (see Figure 19-1). Over half of respondents reported understanding app dependencies (54 percent), assessing on-premises versus cloud costs (46 percent), and assessing technical feasibility (45 percent) as the top three cloud migration challenges.

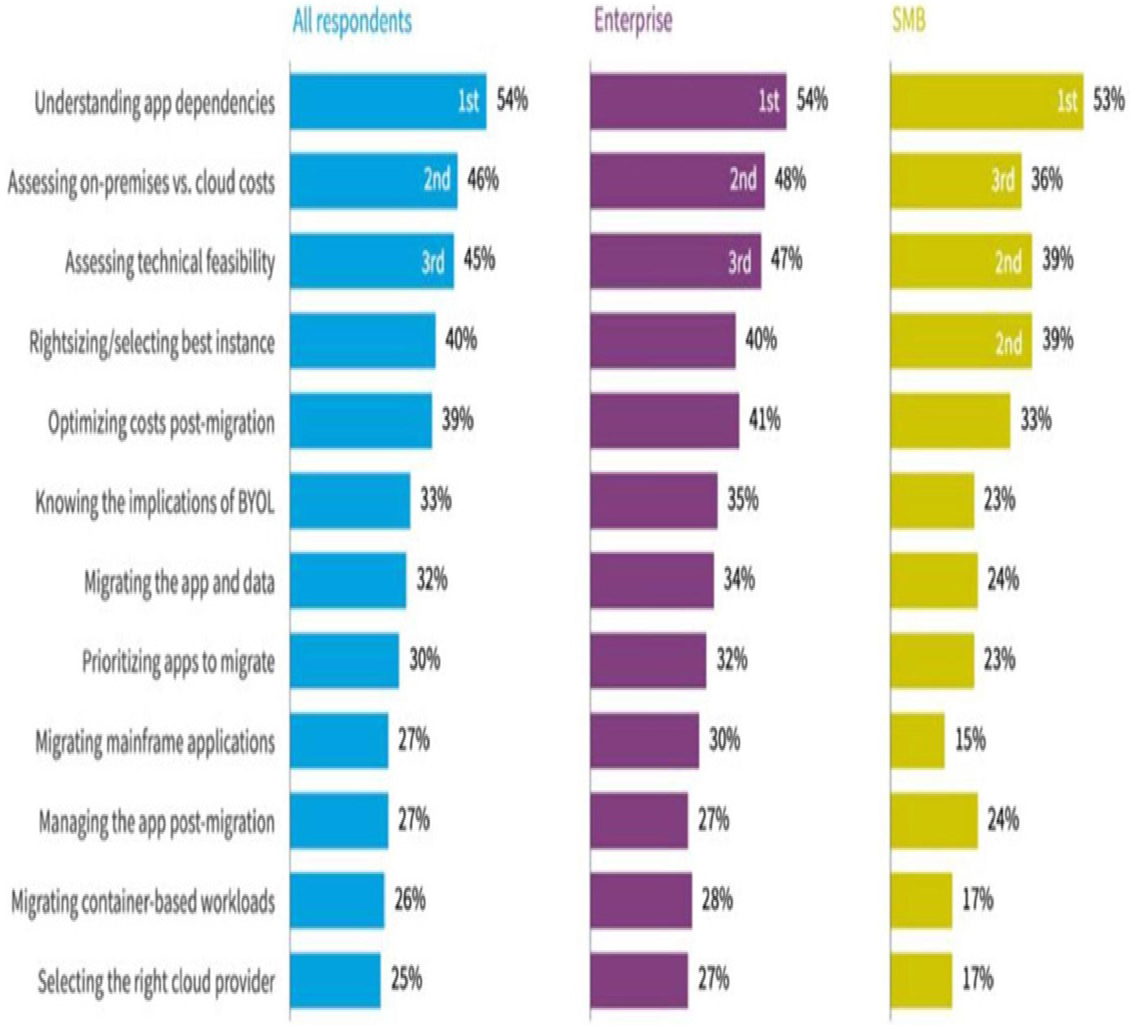


Figure 19-1 *Top Challenges for Cloud Migration (N=752, Enterprise: N=621, SMB: N=132)*

It’s important to minimize downtime in the migration to keep operational continuity. For example, ABC Corp, a mid-sized company with a significant on-premises data center, implemented a phased migration approach. This method was to migrate workloads in phases, rather than all at once. ABC first migrated its noncritical applications to give its team an opportunity to learn the cloud environment. It then migrated customer-facing applications in low-traffic time windows to minimize disruptions. Next, ABC moved its critical databases, making sure that at every stage, proper backups and fallback procedures were present.

Building applications that work together harmoniously and are up-to-date is another big challenge. While ensuring application compatibility so existing applications could fit into the cloud infrastructure, more often than not, they need to be rearchitected to get full use of cloud-native functionalities. As an example, a legacy enterprise resource planning (ERP) system could need to get decomposed into microservices to perform better in the cloud move. This is a challenging and resource-heavy process.

To tackle this challenge, ABC Corp undertook extensive training sessions for their IT team and brought in cloud architects to facilitate the migration. This strategic investment not only guaranteed a smoother transition to the cloud but also established long-term productive cloud management.

Real-World Application: ABC Corp's Cloud Migration Journey

To make these ideas we just went over a little more tangible, let's consider ABC Corp's journey once again. Facing increasing demands for scalability and agility, ABC Corp chose to move some of its workloads to a public cloud platform. The reasoning behind this move was, well, you need to do that as part of better disaster recovery, you want to keep costs in check, and you want to be able to scale up and down based on a volatile customer demand.

ABC Corp faced the following challenges during migration. The company needed a solution that could keep its customer data safe and compliant. Implementing this solution involved creating security layers (encryption, identity management, etc.), and industry-related regulatory compliance (GDPR, etc.). More importantly, ABC Corp wanted to reduce the system downtime. It developed a phased migration strategy to reduce risk. An example included scheduling the migration of customer-facing applications for late at night when there was less traffic, so rollback plans could be implemented if problems emerged.

One of the biggest obstacles was also upgrading those legacy applications —a daunting task that many organizations are still working on. ABC Corp had some monolithic applications that could not be scaled. The company determined to rearchitect those applications into microservices that may be developed, deployed, and scaled independently. This meant analyzing every application’s dependencies and functionality in depth, and making incremental updates to not break any services.

Motivations for Cloud Migration

For those businesses embarking on the journey toward the cloud, and envisioning their strategic business objectives and operational efficiencies, there are a myriad of drivers for their initial move to the cloud. Those motivations can also instead shed light into the bigger trends motivating cloud adoption by various verticals.

Recent trends of cloud workload migration point to work in areas like databases, analytics, DevOps, IoT, AI/ML, and web content as the biggest targets. They can be migrated for better performance, increased security, modernization requirements, and cost savings. But some workloads, such as databases, also consume quite a bit of storage, and thus need cost management and proper planning ahead of time.

Cost Efficiency and Budget Management

There is a lot of talk about the pay-as-you-go nature of cloud platforms, in which the old pattern of class-changing capital expenses on commodity hardware and data centers is eliminated. This is good news for businesses because it allows them to treat infrastructure as OpEx and use it at scale, only when needed, with no waste, nor over- or under-provisioning of resources. A seasonal business, for example, may reduce cloud services during the off-peak season, thus saving costs.

Innovation-Friendly and Fast Development Environment

With instant, on-demand provisioning of resources that scale out or back with changing requirements, cloud solutions drive rapid business innovation and application development. This flexibility helps organizations **deploy new applications and services faster**, fueling growth and innovation.

For example, a **tech company** can host both development and testing environments in the cloud, significantly reducing release times. This accelerated workflow allows the company to **reallocate resources** and

focus more on strategic innovation, driving competitive advantage and business growth.

Access to Advanced Computing Capabilities

The advanced capabilities that cloud platforms offer are high-performance processing storage, and networking. The cloud, therefore, enables compute/CPU-intensive workloads to run efficiently (using AI/ML, big data analytics, as well as IoT applications). For example, a retail store can use on-the-cloud AI to analyze shopping information to create customized shopping experiences instantly.

Enhanced Security and Compliance

Enterprise-level security processes and compliance frameworks are not easy to obtain for an average business, which is why major cloud providers spend hundreds of billions on such security mechanisms. When the cloud delivers better data security, it also directly impacts business compliance with required regulatory standards. The utilization of cloud computing and embedded security enables organizations to gain access to advanced security features like automated threat detection, encryption, and real-time monitoring that safeguard sensitive data. Compliance with various regulatory standards is necessary for organizations, so they also have compliance certifications for General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI DSS), and the like.

This point ties directly into the principles of zero trust, which dictates that every user and device is verified, thus reducing the attack surface and enforcing the principle of least access. All tools used by the cloud providers (which we will discuss detail in the chapter) are central to zero trust architectures. As an example, a financial institution provisioning to the cloud can utilize these capabilities to protect private customer information. Features like advanced encryption and zero trust network access (ZTNA) ensure secure communication and access, while real-time threat intelligence helps prevent breaches.

Focus on Core Business Activities

Since cloud providers handle the management of the IT infrastructure for a company, management can be focused much more on their core business. When businesses are shifted to focus on innovation and development, they can direct more resources in pursuit of competitive advantage. For instance, a biotech company could focus on science rather than IT.

Global Reach and Market Expansion

Another benefit of the cloud lies in the ability of organizations to quickly expand into new locales because the cloud is supported by a worldwide network of data centers, thus improving user experience by deploying services closer to customers and reducing latency and distance between users. For example, a streaming service can utilize cloud data centers around the planet to offer its content at the highest quality with the smallest amount of buffering.

Hybrid IT Infrastructure and Workload Placement

Businesses are adopting a cloud innovative approach with on-premises, multicloud, and co-location solutions. The hybrid IT infrastructure landscape has matured so that it is widely used for combining these solutions. Ultimately, how these environments are managed depends on where an application needs to live, diversification, total cost of operation, performance requirements of the application, and ultimately, degree of control over the data and platforms.

On-Premises Infrastructure

An on-premises infrastructure allows complete control of hardware and data to be kept, with such security and compliance that is at very high levels, which is especially important for sensitive data or industry sectors with high regulatory requirements. The setup also optimizes low-latency networks where data and applications can be accessed immediately. Yet, on-premises solutions have high upfront costs, require in-house expertise, and are not as flexible when it comes to scaling. For instance, a big bank would keep its most critical banking systems in-house for regulatory purposes and control reasons.

Multicloud Strategy

A multicloud strategy ensures that flexibility is provided by multiple cloud providers, while the risk is mitigated by the removal of lock-in and by the avoidance of single cloud provider dependency. It enables enterprises to both minimize costs and optimize workload performance by leveraging leading services from multiple technology vendors. On the other hand, multicloud can forestall management challenges, so solid governance and integration approaches are essential. For example, a worldwide retailer could utilize Amazon Web Services (AWS) for hosting an e-commerce portal, Google Cloud for analytics, and Microsoft Azure for office productivity tools.

Co-Location and Its Role in Modern IT Strategies

Co-location provides key technology necessities while becoming steadily more relevant in digital business domains that need interoperability with cloud services and digital ecosystems strategy. This service improves physical security and operational expense, and provides value over in-house data center management. That makes it really well suited for core workloads where low-latency interconnections are a must. An example of co-location would be the hosting of AI startup machine-learning models near public cloud providers for lower latency and better performance. According to the 2023 State of the Data Center Report, co-location has become increasingly essential in underpinning the digital business environment, which depends on resilient connectivity with cloud services and digital ecosystems. Some of the business drivers for co-location include

- **The Move to Co-location:** A steady trend of co-location and public cloud services is moving businesses away from on-premises data centers. A vital driver behind this change is the operational expense advantages of in-house expertise for the data center’s proper management, which makes many businesses look at co-location as a way of reducing IT spending.

- **Hybrid IT Models and Co-location:** Many businesses have been transitioning toward a hybrid cloud model, which means many co-location data centers are already prime contenders in providing interconnectivity with leading clouds. This point is important for IT decision-makers who want to enjoy public cloud benefits while also containing potential liabilities relating to costs and performance.

- **Benefits of Co-location:** Co-location is scalable and allows organizations to reduce capital expenditures, because they can purchase additional resources as required, with little or no capital expenditure on new infrastructure. It also connects directly with public cloud services, offering high performance at a lower cost.

- **The Growing Importance of Co-location for Workloads:** Organizations are considering co-location as a host to their critical workloads, including artificial intelligence (AI), machine learning (ML), Internet of Things (IoT), and business intelligence (BI), driven by the necessity to lower latency interconnections and management of workloads efficiently over a range of platforms.

- **Security and Compliance within Co-location Centers:** The strength of security and physical protection that co-location data centers offer, combined with the stringent compliance requirements to be met during construction, makes the environment particularly good for sensitive data handling.

- **Co-location for Future-Ready Solutions:** Companies have begun to realize that co-location is an ideal strategy for larger and smaller businesses. It refocuses energy on what is of prime importance—innovation and revenue-generating initiatives—while simultaneously moving away from the complexities and costs of running data center infrastructures. Co-location has become a business-critical approach for modernizing IT infrastructure, given the current rise in security and digital transformation, and it has therefore become a trend for hosting various workloads.

The Deployment of Cloud Smart on Hybrid IT Environments

The strategic implementation of cloud technologies compared to the "Cloud First" strategy is called "Cloud Smart." Where "Cloud First" promotes cloud solutions of any kind or type (prioritizing cloud solutions regardless of the specific use case), "Cloud Smart" asks us to consider the most effective workload environment and cost-performance-security-operational efficiency perspective. This leads to a hybrid IT configuration combining on-premises, multicloud, and co-location deployments.

For example, an organization might keep its financial databases on-premises for maximum control and security; use co-location for performance-critical applications requiring low-latency connections; and leverage public cloud services for development and testing environments, identity and access management (IAM), and backup workloads. This diversified approach ensures each workload is placed in the most suitable environment, optimizing cost, performance, and security.

Cloud Smart Environment for Workload Placement

Cloud Smart respects the relative advantages of on-premises, multicloud, and co-location environments and then builds on that perspective. Here are some of the factors may determine workload placement:

- **Total Cost of Operation (TCO):** TCO covers both the direct and indirect costs of running workloads in an environment. This requires analysis of operational costs compared to capital costs over time and how that affects the overall budget.
- **Application Performance Needs:** Different applications are used in different locations as per the performance needs. Databases and AI/ML workloads may be migrated to the cloud to reduce costs and modernize, whereas latency-sensitive applications may be preserved in co-location for high performance and control.
- **Data and Platform Control:** The issue is more about having most of the control of your data while outsourcing the control over the management of your infrastructure. Co-location data centers combine

the scalability and flexibility of the cloud with the control and security needed for sensitive or mission-critical applications.

- **Strategic Workload Placement:** Using Cloud Smart, more intelligence is applied in deciding which workloads—databases and analytic workloads—usually move to the cloud to capitalize on cost advantages and modernization opportunities. On the other hand, databases come with a high storage demand and need vigilant cost management. For core apps that need speed, co-location is the preferred choice. Development/test environments, identity and access management, and backup workloads are typically deployed in public clouds.

Choosing a Cloud Model with Zero Trust as the Goal

The choice of cloud service model—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Containers as a Service (CaaS), Function as a Service (FaaS)—is much more than an exercise in operational or technical expedience. Although the decision should reflect the technical capabilities and operational goals of an organization as well as the requirements of specific workloads, the decision also needs to be viewed from a zero trust architecture perspective.

Different cloud service models offer different advantages and disadvantages in terms of the level of control you have versus scalability, ease of use, and other technical requirements. Nevertheless, each brings unique security concerns and limits on control. Observing these boundaries means clarifying where an organization’s responsibility ends and a cloud provider’s begins—crucial in observing the zero trust principle of “never trust, always verify.”

Choosing the right workloads to use in a shared hybrid cloud environment requires consideration of an organization’s overall security posture, its ability to enforce consistent security policy, and the needs of individual workloads. A zero trust approach emphasizes reducing the attack surface and protecting access at every layer of the stack, be it infrastructure-, platform-, or application-based. This is where each of the service models intersects zero trust principles.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) gives you the most control you can get over the infrastructure—a double-edged sword in the zero trust context. This control allows for tighter security controls, such as network segregation and identity-based access controls, but it also vastly increases the potential for misconfiguration. Adversaries can exploit, for instance, open S3 buckets or permissive identity roles.

To avoid these risks in an IaaS environment, organizations should implement network layer least privilege by enforcing microsegmentation to isolate workloads and restricting administrative access through IAM policies that limit privileges to the minimum necessary. Additionally, continuous monitoring and automated threat detection solutions should be leveraged to identify and remediate misconfigurations in real time.

On a user-controlled level, IaaS solutions like AWS EC2 and Azure Virtual Machines offer ample flexibility and control, but they require users to actively implement and maintain zero trust principles.

Organizations should implement the following steps to minimize these risks:

- Enforce network layer least privilege using microsegmentation to isolate workloads and minimize lateral movement.
- Restrict administrative access by applying Identity and Access Management (IAM) policies that enforce least-privilege principles.
- Leverage continuous monitoring and automated threat detection to identify and remediate misconfigurations in real-time.

On a user-controlled level, IaaS solutions like AWS EC2 and Azure Virtual Machines offer significant flexibility and control. However, they require users to implement and maintain zero trust principles actively.

Platform as a Service (PaaS)

Platform as a Service abstracts a lot of the underlying infrastructure, leaving developers with an easier path to developing and deploying their apps. Still, like all abstractions, this has its trade-offs in that organizations might find it harder to enforce security policies directly on the platform. In PaaS scenarios, you need vendor-specific controls to make the zero trust work. To align the principles of PaaS to zero trust, organizations must control access to the platform by enforcing identity-based access for developers as well as applications that reach the platform. Corporate zero trust checks over these applications include code reviews and vulnerability scanning to ensure that they are built in a secure manner. Organizations must also assess the security infrastructure of the vendor and see how it aligns with the current zero trust architecture. Now, while nice for rapid development, AWS Elastic Beanstalk and Azure App Service still have a shared responsibility, and you have to manage their access.

Software as a Service (SaaS)

Software as a Service is the most abstracted cloud model because it offloads most security responsibilities to the service provider. However, SaaS also poses unique risks related to data ownership, third-party trust, and integration. To deploy zero trust in an SaaS environment, organizations should enable identity federation and multifactor authentication (MFA) to ensure only authenticated and authorized users can access the service. Data loss prevention (DLP) tools should be used to monitor and protect sensitive data in motion between various SaaS applications. Additionally, user and entity behavior analytics (UEBA) tools can continuously track user behavior to identify any deviations from normal activity patterns. For example, if an employee who typically accesses HR systems during business hours suddenly attempts to download large amounts of sensitive financial data from an unknown device at 2 a.m., UEBA would flag this activity as suspicious and trigger an alert for investigation. Widely used SaaS applications such as Google Workspace and Microsoft 365 provide friction-free functionality for daily business operations, but they also require strict zero trust policies to mitigate risks of unauthorized access and data breaches.

Containers as a Service (CaaS)

For organizations deploying cloud-native, microservices-based architectures, Containers as a Service performs well. Containers are ephemeral and run in a distributed fashion, making it difficult to implement zero trust because this often requires consistent security policies to be in place. For CaaS to align with zero trust, organizations must implement mutual Transport Layer Security (TLS) authentication between containers that require an ID for authentication, which can be accomplished with service mesh (for example, Istio). Container workloads should also be monitored for vulnerabilities or anomalous behavior using runtime security tools. Network policies and ingress/egress controls should only allow the authorized endpoints to connect with one another in containers. While there are many CaaS platforms (such as **Google Kubernetes Engine (GKE)** and **Azure Kubernetes Service (AKS)** that can semi-automate the deployment of containerized applications, a **zero trust architecture** must effectively isolate and secure every container workload to ensure end-to-end security.

Function as a Service (FaaS)

Function as a Service is an extension of the abstraction idea, in that it allows developers to implement functions without doing server management. This makes deploying software easier but also adds complexity to its security, including visibility holes, execution limitations, and third-party dependencies. But organizations need IAM policies that delineate at a function level to adhere to zero trust in the FaaS environments. Event filtering and access controls must be used to ensure functions are triggered only by authorized events. Also, if a function's intended behavior at runtime is deviated from, that behavior needs to be detected in a timely manner. Solutions such as AWS Lambda and Azure Functions represent an ideal asynchronous event-driven application, but the design around function permissions and the manner in which functions communicate with each other need careful consideration to enforce zero trust.

Analysis of TCO and ROI for Workload Migration

A thorough total cost of ownership (TCO) and return on investment (ROI) analysis is key to understand the financial impact of workload migration to the cloud. This analysis needs to consider the costs of cloud services not only directly but also costs connected to migration and training team, along with downtime costs. As an example, when ABC Corp migrated to the cloud, the company considered both the short-term costs of cloud subscriptions and the long-term savings from less maintenance or upgrades to physical infrastructure. It also took into account the costs related to training its IT personnel and the loss of productivity that the initial transition tends to cause.

The ROI factors should also consider the enhanced efficiency, the better scalability advantages, and the bigger revenue growth potential that may come along with cloud migration and thus their contributions to the ROI factors as well. As a result, ABC Corp decided to move to the cloud, and this helped the company scale up its resources on demand as needed to meet customer needs, which led to improved customer satisfaction and, ultimately, increased sales. The analysis needs to map to the enterprise-level strategy focusing on agility, scalability, and innovation.

Migration Context

Flexera's 2024 State of the Cloud Report found that the IT landscape is changing, with 89 percent of organizations adopting multicloud strategy and 73 percent preferring hybrid cloud approach (see the "References" section at the end of this chapter). The emphasis has moved away from the early adoption of cloud technologies and into the placement of workloads in the right place. The objective of this change is better scalability, security, cost management, and data management. For this reason, when conducting a cloud migration analysis, organizations should be asking themselves the following:

1. What unique business goals are leading the migration?

2. What implications does this migration have for existing operations and the customer experience?
3. How much would it cost to migrate, and what will the benefits be?

Moving Applications and Data

During a migration of applications and data, the goal is creating the least amount of disruption to customers and users, of course. How the business problem is solved should inform the migration strategy and preserve or improve competitive differentiation. For example, ABC Corp phased its migration by first planning to move noncritical applications to verify and expand into customer-facing and critical applications during low-use times. By incorporating a phased implementation strategy, ABC was able to reduce risk and reduce impact.

Deployment Model

Choosing the right platform for each application is critical. Considerations include criticalness of business functions and alignment with the cloud environment. Having a common infrastructure across the platforms can mitigate the migration challenges and provide better flexibility for workloads placement. As an example, ABC Corp adopted a hybrid model by keeping core applications (high security) in-house and moving scalable applications to the cloud. This approach enabled ABC to optimize performance, security, and cost.

People, Processes, and Technology

Effectively managing the relationship between **people, processes, and technology** can determine the success or failure of a migration. The key goals are **agility, scalability, and cost efficiency**, all while ensuring **compliance** and **a growing focus on sustainability**.

Another critical factor is **managing technical debt**, which refers to the hidden cost of future rework that arises from choosing a quick, limited solution instead of a more robust but time-consuming approach. **Functional**

and operational impact assessment tools can evaluate specific workloads and help create a **strong migration strategy**.

For example, **ABC Corp** implemented cloud migration tools and services in its source domain to automate **key transition phases** of the migration process, improving efficiency and reducing manual errors.

The pre-migration planning process is highly sophisticated, requiring a deep understanding of both the organization’s existing IT landscape and—the interesting part—its business objectives and expected workloads. Assessing workloads, selecting the appropriate cloud model, and evaluating financial implications form the foundation of effective migration strategies.

However, the urge to modernize and transfer workloads to the cloud can result in failure if proper pre-work and planning are not conducted beforehand, turning the process into an exercise in futility.

Through careful planning for a cloud adoption strategy and migration to the cloud, businesses like ABC Corp can realize efficiency gains at scale as well as new potential for innovation, resulting in a much brighter future in their business and growth roadmap.

Building Out a Secure Migration Plan

Crafting a secure migration plan is not one-size-fits-all; it can be analogous to navigating a precious piece of cargo across choppy waters. Clearly, the goal is to successfully migrate your workloads from the safe havens of on-premises infrastructure to the uncertain commodity cloud environment. This journey, while ripe with opportunities, is fraught with potential perils that must be meticulously navigated to ensure the security and integrity of your digital assets. In this context, we’ll continue to explore the journey of fictitious company ABC Corp. as it undertakes this critical voyage.

Initially, ABC understood that cloud computing proved beneficial in making the company more agile, scalable, and competitive. However, the leadership team knew just how risky it was to migrate sensitive workloads to the cloud. The approach the team created could become a holistic roadmap for other organizations about to head off on a similar journey.

Migration Strategies: From the Five to Seven R's

In 2010, Gartner published the *Five R's migration strategy* (Rehost, Refactor, Rearchitect, Rebuild, and Replace), and cloud migration strategies received additional traction with R models. Over time, cloud computing matured and organizations became smarter about how to approach migration. Then AWS eventually created the 7 R's model, as illustrated in Figure 19-2.

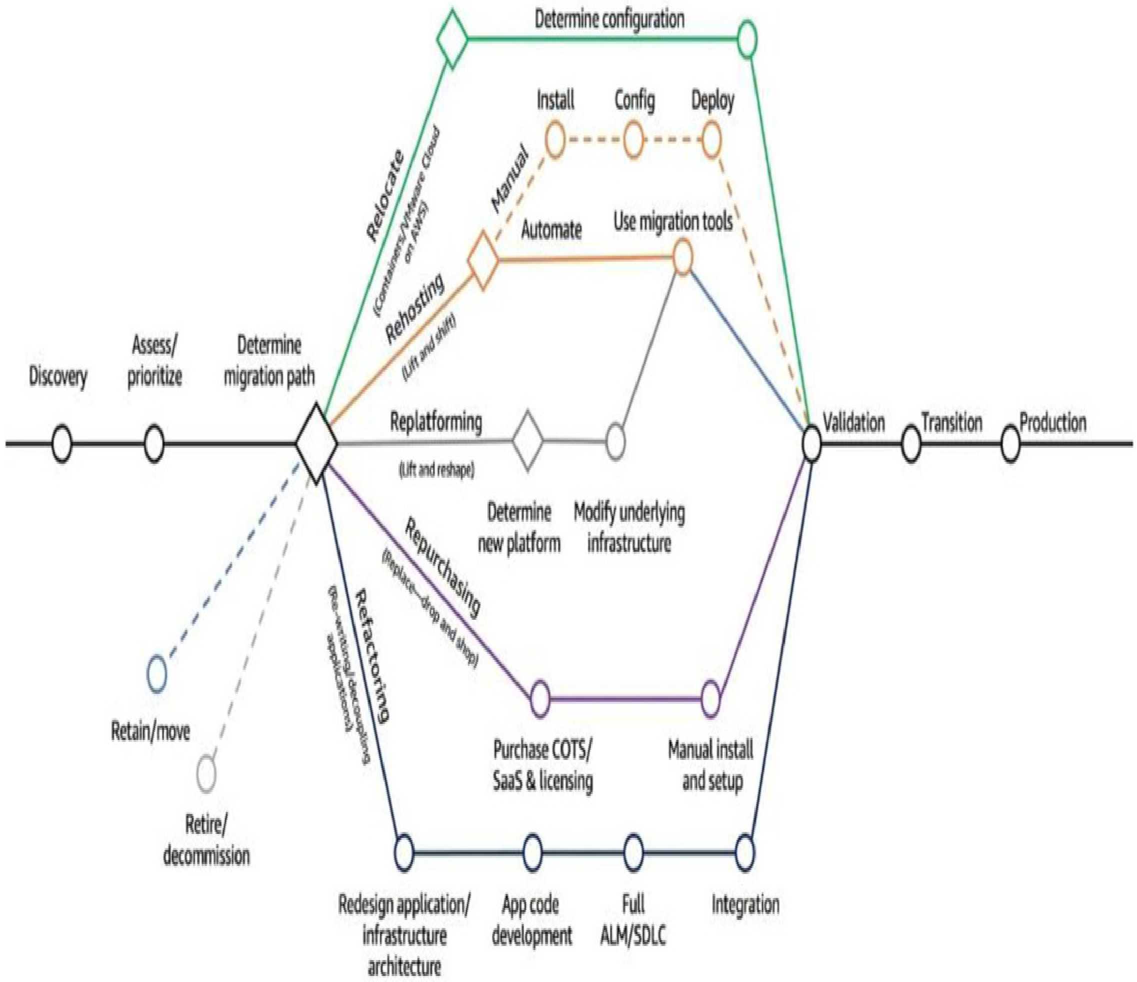


Figure 19-2 *Seven AWS Strategies for Migrating to the Cloud*

With this broader view, we can promote a new wave of thinking about migration, making it clear that some workloads **do not belong in the cloud** or **should be eliminated altogether**. For that reason, let's break down each type of migration strategy in terms of **AWS tools as examples** and incorporate other vendor tools to ensure broader applicability. We will examine the **advantages and disadvantages** of each strategy so that companies like **ABC Corp** have a clear picture of their migration options. ABC evaluated its portfolio to determine the most suitable migration strategy for each workload by using the Seven R's:

1. Rehost (Lift and Shift)

   This strategy involves lifting and shifting applications to the cloud with minimal changes by using tools such as AWS Application Migration Service along with Azure Migrate to help automate this process.

   - Pros: Quick migration with minimal disruption.
   - Cons: May not leverage full cloud optimizations, potentially leading to higher costs in the long run.
   - Some of ABC's applications were moved to the cloud without changes. This strategy was fastest but didn't fully exploit the cloud's benefits.

2. **Relocate (Hypervisor-Level Lift and Shift)**

   This strategy involves transferring applications to a cloud-based version of the same platform, such as from VMware to VMware Cloud on AWS. This strategy minimizes changes to the architecture.

   - Pros: Quick and minimizes disruption.
   - Cons: Limited to certain platforms and may not fully leverage cloud-native features.

3. **Repurchase (Drop and Shop)**

   This strategy involves migrating to another software version or cloud-native (SaaS model) product, such as going from an **on-premises Customer Relationship Management (CRM)** system to **Salesforce SaaS**.

   - Pros: Access to new features and reduced maintenance.

- Cons: Can be costly and requires adapting to a new platform.

ABC decided to rebuild some legacy systems from scratch using cloud-native technologies, prioritizing long-term benefits over short-term efforts. Where it was more efficient to implement Software as a Service, ABC used SaaS to eliminate the overhead of maintenance by replacing existing applications.

4. **Replatform (Lift, Tinker, and Shift)**

   This strategy involves doing lightweight migrations to benefit from the cloud features without completely rearchitecting, such as moving a database to Amazon Relational Database Service (RDS).

   - Pros: Balances quick migration with some cloud benefits.
   - Cons: Limited optimizations may miss out on deeper cloud-native advantages.

   It was essential for ABC to leverage cloud-native features without a complete rebuild. This involved minor modifications to the application code.

5. **Refactor (Rearchitect)**

   This strategy involves redesigning applications to be cloud-native to maximize agility, performance, and scalability, utilizing services like AWS Lambda for serverless architectures.

   - Pros: Fully leverages cloud benefits for scalability and performance.
   - Cons: Most time-consuming and complex strategy, requiring significant investment.

   More complex applications have been rearchitected to become true cloud-native apps with better scalability and performance.

   Although the migration strategies were predominantly centered around the Five R’s (Rehost, Refactor, Rearchitect, Rebuild, Replace), the idea has evolved to encompass two further strategies that provide detailed insights on the options of migration that an organization has: Retire and Retain.

6. **Retire (Stop Using)**

   This strategy involves decommissioning applications that no longer provide business value or pose unnecessary costs and security risks. Tools like AWS CloudFormation can help identify and manage resources for decommissioning.

   - Pros: Reduces costs and focuses resources on valuable applications.
   - Cons: May require thorough analysis to avoid retiring applications with hidden dependencies.

   Part of the migration process involves identifying obsolete or redundant applications that can be decommissioned, reducing complexity and cost.

7. **Retain (Revisit)**

   This strategy involves opting to keep certain applications on-premises due to compliance, high risk, or other strategic reasons. AWS's extensive documentation provides guidance on assessing which applications to retain.

   - Pros: Maintains compliance and manages risk effectively.
   - Cons: Could delay achieving full cloud benefits for certain workloads.

   In some cases, compliance, latency, or other strategic imperatives could also drive a need to retain certain applications or workloads on-premises for the betterment of the organization.

Fueled by these approaches, enterprises must craft their own path to cloud migration to balance their business initiative, technology need, and risk appetite. Strategy decisions should be driven by the assessment results of a complete application portfolio that are weighted across application complexity, importance to the business, and cloud readiness.

Security Considerations for Data Transfer and Transition Phases

The transition phase was a critical juncture where data was most vulnerable—perhaps the riskiest part of the migration process, because it involves the active migration of data, applications, and workloads from on-premises infrastructure to the cloud. During this phase, ABC Corp used secure modes of transferring data such as VPNs and Direct Connect to ensure data integrity and confidentiality; plus, it set up rollback procedures to minimize disruption and maintain business continuity in the event that unexpected problems did occur.

The transition phase is more than a technical challenge; it is a strategic inflection point, with significant risk management implications including data breaches, misconfigurations, and operational downtime. Security policy migrations that are out of sync or corrupted data while migrating can have deep and lasting ramifications and undermine the success of the entire effort. Through careful planning, proactive risk management, and stringent security measures, organizations such as ABC Corp can lay the groundwork for a secure and high-performing cloud environment that meets operational and strategic business requirements.

This secure approach allowed ABC to navigate the complexities of cloud migration effectively, underscoring the importance of a well-thought-out strategy that balances the benefits of cloud computing with the imperatives of security and business continuity. The secure migration strategy is really beyond just migrating workloads per se and more about migrating the operational, security, and business landscape of the organization. Each step of the workload mobility process is a building block to a secure, agile, and efficient cloud, which we will explore further as we go through subsequent steps.

Integrating AWS's Well-Architected Framework: Case Study of ABC Corp

With principles and best practices of the AWS Well-Architected Framework implemented in ABC's secure migration strategy, the steps get amplified for security and resilience in the company's cloud journey. Through guidance from AWS, ABC will ensure a robust and secure cloud architecture under industry best practices. Let's discuss how these principles and practices can be applied practically to ABC's migration strategy.

Building a Strong Identity Foundation

ABC builds a strong foundation of identity because, without it, the cloud cannot be secured. ABC follows the principle of least privilege. In that case, with people and systems, separation of duties is enforced, and static credentials are reduced. Centralization of identity management allows ABC to rationalize access controls and mechanisms of authentication that will affect the entire cloud. In this way, ABC minimizes the attack surface.

Maintaining Traceability

Maintaining traceability is important for security and compliance. ABC includes the following user activities and changes in the environment in real time by using intensive monitoring and alerting systems. By automatically integrating the logs and metrics in the system of automated investigation and response, ABC can quickly identify and thwart future threats that may be posed to the system. This approach augments not only security but also gives critical insights into performance on the operational level.

Applying Security at All Layers

Following the defense-in-depth approach, ABC applies multiple security controls to all levels of the cloud architecture. Security controls have been implemented from the perimeter network down to the application and code levels in each layer. They include network firewalls, intrusion detection systems, data encryption, and secure coding practices for that layer. By layering security defenses this way, ABC ensures that if one control fails, there are others in place that will continue to protect.

Making Security Best Practices into Habits

Automation is one of the cornerstones of ABC’s security strategy. Security configurations and controls are automated everywhere possible to allow the quick and efficient scaling of security measures. Security as Code allows for the use of predefined, versioned security templates that can be deployed across the cloud environment, so human error is minimized in a way that makes the enforcement of consistent security policies possible.

Ensuring Data Is Protected in Transit and at Rest

ABC classifies data based on its sensitivity and then applies stringent controls to protect this data both in transit and at rest. The company uses data encryption, tokenization, and strict access controls to secure data against unauthorized users and breaches. This is quite important now because the company has susceptible information.

Keeping People Away from Data

Keeping people away from data makes it possible for ABC to use a more minor degree of manual access to data and reduce dependence, minimizing the chances of data being exposed or mishandled. In this regard, automated systems are put in place for data processing and management through which sensitive data goes; this approach significantly minimizes human errors and upholds data integrity.

Preparing for Security Incidents

Acknowledging the fact that incidents can and do occur, ABC has already come up with well-documented end-to-end incident response plans and policies. The team’s preparedness is shown through regular incident response drills and simulations, which are efficient in the management and mitigation of security events. Automation tools help enhance detection, investigation, and recovery time from security incidents.

Performing Risk Assessment and Mitigation

ABC realized that security risks would be at the top of its list of issues. The company undertook rigorous risk assessments focusing on breaches of data, loss of control over data, and compliance challenges. Mitigation strategies focused on encrypting data in transit and at rest, using rigorous access controls, and selecting cloud providers compliant with industry standards.

Embracing the AWS Shared Responsibility Model

ABC adheres to the AWS Shared Responsibility Model, which explains how security responsibilities are shared between AWS and its customers. By accepting that securing the infrastructure is an AWS responsibility, ABC can put all its effort into securing workloads and data throughout the AWS environment. This framework allows ABC to continue benefiting from innovations in security and automation by AWS toward better protection.

Connecting with the Well-Architected Frameworks

The Well-Architected Frameworks of all major cloud service providers, such as AWS, Google Cloud, and more, articulate best practices and core strategies of cloud architecture around operational excellence, security, reliability, performance efficiency, and cost optimization in ensuring the design and operation of reliable, secure, efficient, and cost-effective systems within the cloud environment. These frameworks provide guidelines and leading information for constructing cloud architecture with a focus on design and operation of reliable, secure, and cost-efficient systems.

In fact, the approach ABC takes to develop a secure migration strategy to the last detail naturally maps into the pillars of the Well-Architected Frameworks. It ensures that each application and workload is fit for deployment into the cloud and then maps the appropriate migration strategy to make sure that the cloud environment is not merely a reflection of the on-premises setup but is optimized, secure, and efficient, employing cloud-native features.

1. **Operational Excellence:** The strategy of migration for ABC to include automation tools and CI/CD pipelines will ensure faster operations of the cloud.
2. **Security:** Concerning data protection and compliance, this is the pillar where risk assessment, data encryption, and methods for transferring secure data have a direct contribution.
3. **Reliability:** The ABC strategy also serves to maximize the reliability of its deployment through high availability (HA) planning, disaster recovery, and the establishment of rollback procedures.
4. **Performance Efficiency:** To rearchitect such applications for cloud-native capabilities speaks to the performance efficiency pillar that drives optimal use of the cloud’s resources.
5. **Cost Optimization:** Using TCO and ROI analysis and with the proper selection of migration strategy (for example, Rehost, Refactor), ABC ensures that its migration is cost-effective and, to that end, aligned with the cost optimization pillar.

The need, therefore, to work with Well-Architected Frameworks becomes real; the purpose is to ensure that success is achieved in the migration process and that the approach chosen has the assurance of sustainability with efficiency for all operations within the cloud.

The secure migration strategy pursued by ABC, inspired by such resources as this presentation, is rather holistic and is an approach that emphasizes integration with established frameworks, adaptability to the ever-changing landscape of cloud computing, and maintenance of organization resilience, security, and agility.

Workload Migration Frameworks and Tools

Transitioning organizations from on-premises data centers to cloud environments is not only about moving where you host the data or applications. A change in the way business would secure and manage its IT assets and a workload mobility–oriented, transformational journey. This requires detailed planning and execution, in addition to practical tools and services.

Leveraging Migration Services and Tools

A variety of tools are available from cloud service providers (CSPs) and third-party vendors to assist in transitioning. AWS Migration Hub, Azure Migrate, and Google Cloud's Migration Center offer central platforms for evaluating, planning, and tracking your migration; each one is geared toward different organizational objectives. These tools also include several migration strategies, such as rehosting, refactoring, rearchitecting, rebuilding, or replacing.

Following are some examples of key hyperscalers that use cloud platform migration tooling:

- **Azure Migrate: A Centralized Migration Solution:** Azure Migrate is a full-funnel solution that provides assessment and migration tools for on-premises servers, databases, applications, and data. It has a single dashboard to keep track of all the moving parts of the migration process, all in one place, and at your disposal so that nothing gets out of hand. Azure Migrate also helps to assess the on-premises environment with tools like Server Assessment and migrate workloads seamlessly to Azure with tools like Server Migration to facilitate factors like compatibility and optimum sizing.
- **AWS Migration Hub: Streamlining the AWS Journey:** AWS Migration Hub offers an integrated solution that guides organizations through the migration and modernization process on AWS. It streamlines the entire process of discovery, assessment, planning, and execution, with automated recommendations as well as a single pane of glass tracking to give an overview of the entire migration.

AWS Migration Hub gets you out the door faster with an established process and features to minimize manual tasks and enable smoother migration. It is capable of orchestrating complex migrations and supports diverse strategies like rehosting, refactoring, and replatforming, ensuring organizations follow the correct path for their specific requirements.

- **Google Cloud Migration Center:** Google Cloud Migration Center provides flexible data and application migration capabilities powered by AI to simplify and accelerate application migration and modernization. Google Cloud Platform (GCP) provides a complete set of tools to support the assessment, planning, and execution of transitions to GCP efficiently and securely.

Understanding the Role of Automation and Integration

Migrating workloads to Google Cloud with automation increases efficiency, accuracy, and security. This ensures consistency and compliance because Infrastructure as Code (IaC) tools like Terraform and Ansible can be used to automatically provision your cloud resources in the future. Using this approach, you don’t have islands of controls where each infrastructure is controlled with inconsistent and untracked methods. Furthermore, the inclusion of continuous integration and continuous delivery (CI/CD) pipelines into the migration activity encourages a DevOps model that supports iterative testing and deployment, which, in turn, reduces errors while fast-tracking application readiness for cloud environments.

Ensuring Data Security During Migration

Ensuring data security during migration is a top priority. Data in transit is secured using VPNs and Direct Connect’s secure transfer methods, which help maintain data integrity and confidentiality.

Compliance with industry regulations and geolocation laws must be maintained throughout the migration process. To prevent security breaches, sensitive data must be protected during and after re-platforming through encryption at rest and fine-grained access controls. These measures ensure that security policies are enforced dynamically and automatically, reducing risk in real time.

Optimizing Cloud Migration Outcomes

The success of cloud migrations typically relies on strategically choosing the right frameworks and tools. This effort contributes to security, effectiveness, and operational excellence. Learning about new tools and best practices will be key to aligning your migration strategy with these objectives. As cloud technologies evolve, so must we.

The "What" to Migrate

Proper identification of what components to move where is key to what and how you draw your path and steer your cloud migration strategy. This includes

- **Compute Infrastructure:** Virtual machines (VMs), containers, and application servers
- **Storage Infrastructure:** Solutions for data storage, archival, and file sharing
- **Networking Components:** Firewalls, load balancers, VPNs, and so on
- **Databases and Analytics:** Tools to ensure zero data loss and optimal performance post-migration
- **Business-Critical Applications:** ERP, CRM, and other line-of-business domain-specific software
- **Middleware:** Tools that connect layers between different parts of IT infrastructure, like enterprise service buses (ESBs) and application programming interface (API) gateways
- **Cloud Identity Systems:** Tools to manage access and identity in the cloud

- **Management Tools:** Tools for configuration management, monitoring, and operations
- **Developer Infrastructure:** Any code or resources available to provide support for developers (for example, code repositories, testing environments)

Pinpointing these elements enables organizations to prioritize which migrations are major business objectives while also identifying where tooling and best practices can reduce cost impact.

Comprehensive Tooling for Migration

Tools and services used to assist the entire migration process are grouped by their primary types of functions:

1. **Assessment and Planning Tools:** Organizations use these tools to assess their on-premises environment to get an idea about cloud readiness (or suitability) and to determine the best strategy for migration. This includes tools such as Azure Migrate Server Assessment and Discovery and assessment capabilities in AWS Migration Hub.
2. **Migration Services:** These services aid in moving workloads from data centers or other clouds and help with migration strategies such as rehosting, refactoring, rearchitecting, and replatforming. Some of these are visible in Azure Migrate: Server Migration, AWS Database Migration Service, and Google Cloud Migration Center services.
3. **Data Migration Tools:** These tools play a vital role in data migration; they ensure that you transfer your data to the cloud without losing data integrity and have minimal downtime while transferring your data. Some examples of these tools include Carbonite Migrate (a third-party service that uses real-time, byte-level replication to duplicate a source system to the target) and AWS Snowball (a petabyte-scale data transport service that leverages its own secure devices to transfer large amounts of data into and out of the AWS cloud).

4. **Cloud-Based Application Performance Monitoring Tools:** These tools monitor application performance post-migration to ensure the application is functioning optimally in the cloud environment. Application performance management tools such as AppDynamics and the operations suite of Google Cloud offer monitoring and analytics capabilities in real time.
5. **Automation and Orchestration Tools:** Of course, tools like Terraform and Ansible allow Infrastructure as Code to set the stage for provisioning. Orchestration tools help to control the deployment and management of cloudwide apps and workloads.
6. **Security and Compliance Tools:** These tools help protect data and applications during migration and after migration and ensure they meet compliance and regulatory standards. These features comprise data encryption, identity and access management for granular access control, and continuous security monitoring. For example, if the data is moved to a service such as Amazon S3, IAM policies can be used to control exactly who has access to which objects. Similarly, these principles apply across other cloud storage solutions and services, ensuring robust protection, regardless of the chosen destination.
7. **Cost Management and Optimization Tools:** To control and optimize cloud spending, these tools provide insights into resource utilization and cost-efficiency, helping organizations optimize their cloud resource allocation and reduce unnecessary expenses.

Data Security During Workload Migration

Think of the feeling you have while moving homes. Like moving a home, migrating workloads to the cloud involves transporting large amounts of data over the Internet. Just like you would not want lost, damaged, or missing personal possessions on your way to a new home, keeping your data safe on the way from servers on the ground to the online cloud (and back again) is a top priority.

Data security during workload migration requires a detailed plan to ensure a smooth transition. It’s about guaranteeing that your digital assets are securely packaged, safely transported, and correctly unpacked in their new cloud environment, all while maintaining privacy.

Here’s the breakdown:

- **Securing the Package:** Prior to the move, it’s crucial to encrypt your data, like wrapping breakable items in protective bubble wrap. Encryption obscures the data, rendering it indecipherable to anyone who might intercept it during the transfer.
- **Safe Transport:** Just as you would hire a reliable moving company with a secure vehicle for a physical move, digital migration involves using secure data transfer methods such as VPNs or Direct Connect. They provide a protected channel for your data to pass through the wide net of the Internet without falling into the hands of cybercriminals.
- **Careful Unpacking:** Your data is not simply dumped out on the cloud when it arrives. It’s cautiously unpacked to ensure its integrity and proper placement, ready for use as if it had never left its original location.

The migration, however, is not the end of the process: You must not forget to tailor the security of the cloud environment and deploy firewalls, access controls, and nonstop monitoring. This way, the data is protected during its transit as well as after it has reached its final destination.

Imagine a house that can adapt its security systems in response to suspicious activity. That’s the advantage of cloud environments; they provide advanced security features capable of real-time responses to potential threats.

Coming back from our household story to the cloud context, security during workload migration involves safeguarding your digital assets from the moment they depart from your on-site environment, through their journey across the Internet, to their final destination in the cloud. This stage is vital because data in this digital age is not just information, but the backbone of businesses, the memory vault, and sometimes the foundation for digital identity. As with any good story, it ends happily: Your data is now safely living in its new cloud mansion.

Ensuring Data Integrity and Confidentiality

What does it mean during workload migration? If that sounds daunting, let’s simplify it. Imagine you are sending an encrypted message to a friend who lives in another city, and you have your mind set on ensuring that

- **Your message is a secret (Confidentiality)**—Confidentiality is like telling a secret and then dropping it in a lockbox where only you and the person you told has a key to that lockbox. So in the digital world, this lockbox is encryption. In other words, when you take your data and put it in the cloud in an encrypted format, you are, in essence, creating a secret code only accessible to an authorized user (an authorized person with the correct decryption key). This means that if your data is hijacked even halfway, only gibberish would be displayed.

- **It arrives to them as you wrote it (Integrity)**—Data integrity, on the other hand, is ensuring that the message cannot be modified on the way. It acts like a letter sent in a sealed envelope. If someone attempts to hack that message, you will detect it. In the digital world, this process is commonly handled using hash functions and checksums. Before your data is transmitted, these methods will produce a digital fingerprint unique to you. When the data reaches its destination in the cloud, the system rechecks this fingerprint. When the fingerprints align, your data gets through unscathed. Otherwise, something is wrong, and some things might have changed or corrupted the data during the transmission.

While keeping the bad guys out is always important, protecting data integrity and confidentiality during workload migration also matters. And it is also about ensuring that the data does not get lost in translation or inadvertently transition into a different form during its digital journey. For businesses, this issue is pertinent because we live in a world where even a minor change or exposure could cause larger adverse consequences; that is driven by data, of course, and all it takes is for the business to make a wrong decision based on corrupted data or sensitive information to fall into the wrong hands.

Therefore, maintaining data integrity and confidentiality is like the best secret-keeper making sure that secrets (data) are not only kept secret but also delivered correctly. It’s a non-negotiable part of migrating workloads to the cloud because, at the end of the day, you want to trust your cloud environment just as much, if not more, than your on-premises setup.

What Is Secure Data Transfer?

Secure data transfer is the transfer of data from one location to another with the higher probability that the data will not be intercepted, altered, or otherwise corrupted in transit. Secure data transfer aims at keeping the information to be transferred safe and sound.

Within the context of cloud migration, data transfer mechanisms are essential for transferring large amounts of data from on-premises to the cloud in a fast and secure manner. It spans a range of offerings from network solutions—like Direct Connect and VPNs—to online and offline transfer methods, storage gateways, and data transfer appliances. All these cloud providers provide their own flavor of it, thus giving businesses flexibility and options while moving their workloads to the cloud.

To understand how organizations can securely transfer data to the cloud, we categorize these methods into **three main approaches**:

- **Online Data Transfer Methods** (transferring data via network connections)
- **Hybrid Data Transfer** (combining online and offline strategies for efficiency)
- **Offline Data Transfer Methods** (moving data physically using appliances)

Types of Cloud Data Transfer

Cloud providers have multiple data transfer services to meet different types of migration requirements. Depending on **data volume, speed, and security needs**, organizations may choose **online, hybrid, or offline** migration strategies.

Cloud providers have multiple data transfer services to meet different types of migration requirements.

Online Data Migration

Online data migration means sending data over the Internet or dedicated network link to the cloud. When you have a continuing requirement for data sync or if the data set is not too large, you can use this approach.

- **Network Data Transfers:** There are various methods of transferring data to the cloud based on the public Internet and dedicated network connections.
- **Database Migration Services:** These migration services are specifically designed to create fully available database transfers with as little downtime as possible.
- **Data Transfer Acceleration:** Some cloud providers offer acceleration services that optimize the transfer of data over long distances, significantly speeding up the migration process.
- **DataSync-Like Services:** These services automate the movement of data between on-premises storage and cloud storage, handling tasks such as encryption, script management, and network optimization to facilitate faster data transfers.

Network Data Transfers

When we talk about cloud migration, network data transfers are basically data that moves to the cloud over an Internet connection or a dedicated network link. This transfer encompasses a variety of methods:

1. **Public Internet Transfers:** This type of transfer involves using the public Internet that is already in place to move data directly to the cloud but without creating further, bespoke networking infrastructure. This is the most common type of online data migration, although the speed available will be limited to the bandwidth of your connection, and it might not always be as safe or fast as you would like.

- It uses standard Internet protocols (HTTP, HTTPS, FTP, SFTP, etc.) for data transfer. Although HTTPS and SFTP encrypt the data during transmission, the base layer is still the public Internet.
- This method is universally accessible, with no additional contracts or setup required above that needed to gain access to cloud services in the normal way. It is accessible by any individual or organization via the Internet with appropriate credentials to the cloud service.

2. **Virtual Private Network:** A VPN extends a private network across a public network, allowing you to send data between your on-premises network and your cloud environment over an encrypted, secure connection. It's a form of online data transfer that enhances security but still relies on the public Internet's bandwidth. While cost-effective and relatively easy to set up, VPNs may not provide the bandwidth necessary for large-scale migrations.
3. **Direct Connect Services:** These services provide dedicated cloud links and are a critical component for large, secure online data migrations and comprehensive hybrid cloud architectures.

Direct Connect Services

Solutions such as AWS Direct Connect, Azure ExpressRoute, and Google Cloud Interconnect offer a private network link for traffic between your on-premises infrastructure and the cloud, circumventing the public Internet. These solutions provide key benefits for strategies that deal with online cloud migration, such as low latency, stable bandwidth, and better security because data will never pass over the public Internet.

Moreover, these solutions usually have lower data egress charges than Internet transfer charges, and larger packet sizes (MTUs), to make the transfer of data more efficient. Because cloud providers have access to global network backbones, this can result in fast data transfers of 1, 10, or even 100 Gbps, and can go lower than 1 Gbps through partnerships.

Configurations like virtual interfaces (VIFs) extend VLANs from the cloud to your premises, enabling private connectivity with your infrastructure, access to public cloud services without the necessity of exposing any of their components to the Internet, and flexible connections to remote regions or multiple virtual private clouds (VPCs) or other cloud resources through transit gateways. These characteristics make Direct Connect solutions one of the most important parts of an efficient, secure, and cheap online cloud migration strategy.

While these services facilitate online data transfer by providing a more stable and high-bandwidth connection, they can also be seen as part of a hybrid approach due to their role in connecting on-premises infrastructure directly to the cloud. Typically, these hybrid cloud services will help to bridge the connection between your existing infrastructure and the cloud. A dedicated VPN connection allows both on-premises plus the cloud to work as a single system.

Hybrid Data Migration

Hybrid data migration uses a combination of cloud and on-premises data migration solutions, which is perfect for scenarios where low-latency access to the data is required or for gradually integrating on-premises legacy systems to the cloud. Storage gateways can be used in addition to Direct Connect services. These connect on-premises applications with cloud storage, enabling low-latency data access and integration. They are useful for diverse needs, including file, volume, and tape (aggregates) gateways, which allow various storage and backup requirements.

Offline Data Migration

Offline data migration means moving data to a cloud provider via shipment of data on devices. This is the best method to use when migrating a large volume of data. Offline data migration is a valid option when large datasets that take a long time to move over the Internet make this movement impractical—if, for example, a cloud service is located remotely or simply has limited Internet availability—or for data that needs to be kept under a tight security protocol while on the move between on-premises to the cloud.

- **Data Transfer Appliances:** Comparable to AWS Snowball, these rugged, encrypted devices are used to move large amounts of data. The choice between smaller devices and large-scale solutions like

AWS Snowmobile (or similar) depends on the volume of data and the urgency of the transfer.

- **Portable Storage Devices:** For edge computing or ultra-portable data transfer needs, devices akin to AWS Snowcone can be deployed, supporting both data transfer and computing capabilities in remote or offline locations.

Considerations for Each Approach

There are many aspects to be considered while defining data migration strategies. The amount and size of data to be migrated play a major role in deciding which sort of process will provide optimal experience—online or offline or hybrid migration. The speed demands are also significant; how quickly the data needs to move will often guide whether network transfers, offline appliances, or direct connections are desired. Security and compliance are key, meaning data must be encrypted in transit as well as meet the standards of regulatory obligations through either route. Cost is an equally essential aspect, including the obvious cost of service and other indirect costs like having to deal with downtime or upgrading bandwidth. Finally, the operational impact should be evaluated, comparing how each of them influences running operations and intending to pick an approach that causes lowest disturbance.

Secure Data Transfer Best Practices

The following are six best practices to assist in securing your company's data transfer:

1. **Formulate Your Data Strategies:** Each business has a unique way of using its data, necessitating a strategy that fits your needs. This involves identifying the specific needs for your data and how you plan to use it. This step will aid in determining file formats, data types, and other parameters to consider during the data transfer process. Also, you need to establish measures to prevent data loss and ensure compliance with data privacy regulations.

2. **Implement Reliable Data Protection Systems and Protocols:** Data protection is vital to your business. Protecting your data during transfers will prevent leakages, breaches, or other potential mishaps. Data encryption methods can help you safeguard your information while in transit.
3. **Secure Information with Access Control:** Securing your data is a primary concern when storing, reading, or sharing corporate data. Only authorized personnel should have access to the data. You need to establish proper authentication methods and permissions for each user.
4. **Implement a Viable Communication Strategy:** Data transfer between systems is a complex process, and communication is critical to its success. Understanding this will aid in determining the data migration order, setting up dependencies, and establishing a viable communication strategy.
5. **Always Have a Backup:** Even with the best data management practices, there's still a chance that you'll lose some of your data. Hence, it is essential to have a backup of your data before starting the migration process.
6. **Comply with Data Protection Laws for International Transfers:** Complying with data privacy legislation for cross-border transfers is critical to securing data transfer from one place to another, especially when the transfers involve the EU and non-EU countries.

Data transfer is indeed a critical business process that requires planning beforehand. A detailed data management plan will assist in preventing data loss and ensuring a smooth transition of information from one system to another with minimal disruption. You need to always prioritize security and communication to prevent disruptions and have a backup plan in case of an emergency.

Data Transfer vs. Cloud Migration: An Overview

Differentiating between the concepts of data transfer and cloud migration is essential for a comprehensive understanding of cloud adoption strategies.

While closely related, these concepts address different stages and challenges within the broader context of moving to the cloud.

- **Data Transfer:** This strategy specifically talks about moving the data from one location to another location—for example, on-premises servers to cloud storage. Although it is just part of an overall strategy, it is fundamental to many organizations for moving to cloud services.
- **Cloud Migration:** Cloud migration is the process of moving data, applications, and IT assets from an on-premises infrastructure to a cloud-based environment—requiring planning, execution, and management to not just transfer data safely and productively but also to optimize and integrate cloud services into an organization's IT ecosystem.

Differentiating between **data transfer** and **cloud migration** is essential for a comprehensive understanding of cloud adoption strategies. While closely related, these concepts address **different stages and challenges** within the broader process of moving to the cloud.

- Data Transfer: This refers specifically to moving data from one location to another, such as transferring data from on-premises servers to cloud storage. Although data transfer is just one part of an overall cloud strategy, it is a foundational component for many organizations migrating to cloud services.
- Cloud Migration: A broader activity that involves moving not just data, but also applications and IT processes from on-premises or legacy infrastructure to the cloud. Cloud migration requires careful planning, execution, and management to ensure data is transferred securely and efficiently, while also optimizing and integrating cloud services into an organization's IT ecosystem.

Data Transfer Considerations as Part of a Broader Cloud Migration Strategy

Cloud migration (as opposed to the transfer of data to the cloud) includes much more than just the transfer of data; it also includes functionalities regarding the integration of applications, workloads, or cloud services, plus their optimization and ongoing management.

The Impact of Bandwidth Limitations

Bandwidth defines the **maximum data transfer rate** through a given path. **In the context of cloud migration, an enterprise's ability to interconnect with a cloud provider determines how efficiently data can be transmitted to the cloud environment.** Low bandwidth results in **slower data transfer speeds, longer migration times, and potential downtime.**

Bandwidth limitations are a significant challenge in the context of cloud migration, affecting the speed and efficiency with which data can be moved from on-premises environments to the cloud. Bandwidth limitations should be understood and addressed properly in a migration so that they do not disrupt business operations but keep costs under control.

The impact of bandwidth limitations on data migration can be substantial. First, constrained bandwidth can slow migration, prolonging the transfer of large data volumes into the cloud and postponing project timelines. Second, the excessive time it takes to transfer data can become expensive if cloud resources are rented temporarily from cloud providers for the migration effort, or new network capacity is acquired temporarily. Third, slow data transfers can become disruptive to operations, leading to prolonged outages or limited functionality in important business applications, and in turn, impacting business operations. Last, if data changes frequently, the complications of keeping data synchronized between on-premises and cloud environments become much more complex with limited bandwidth.

Tactics for Mitigating Bandwidth Limitations

Bandwidth limitations are an important aspect that does pose a challenge in cloud migration activities; however, if planning is done well, including the use of a few smart tactics, the impact of bandwidth limitations can surely be mitigated:

1. **Data Prioritization:** Determine the order in which data and applications are migrated based on business requirements. Data prioritization ensures that the most important data gets lifted and

goes functional in the cloud space even while a full migration is still running its course.

2. **Data Compression:** One of the low-hanging fruits with a data transfer would be to compress the data you want to transfer. Inevitably, moving this much less data requires less bandwidth and reduces its transfer time.
3. **Incremental Migration:** Instead of migrating all data at once, consider an incremental approach. Copy vital information first, then less vital information. This approach can even include delta migrations, such as transferring changes to the data only after the first migration, minimizing the amount of data that needs to be transmitted.
4. **Off-Peak Migration:** Performing data transfer during off-peak hours will ensure more bandwidth is used and have less impact on regular business.
5. **Dedicated Data Transfer Services:** For larger data migrations, make use of the specialized services offered by cloud providers like AWS Snowball, Azure Data Box, or Google Transfer Appliance. These services involve physically shipping data using secure hardware devices, bypassing Internet bandwidth limitations.
6. **Network Optimization:** Implement and use WAN optimization and SD-WAN technologies to efficiently increase the amount of data transferred over existing bandwidth capacity.
7. **Upgrade Bandwidth Capacity:** If feasible, consider upgrading your Internet connection or negotiating temporary bandwidth increases with your service provider for the duration of the migration project.
8. **Hybrid Migration Approaches:** Use a combination of cloud and on-premises solutions during the migration phase to gradually move workloads to the cloud without overwhelming your bandwidth.

Data Security Concerns

Data security is critical in transferring data as well as in cloud migration. Ensuring data privacy and regulation compliance (such as GDPR) and **protection against breaches and cyber threats** must be key considerations during migration. These concerns may be mitigated by utilizing strong encryption standards, data sovereignty practices, and integrated data protection features to create a secure migration path.

Downtime Risks

Avoiding downtime is one of main goals when migrating to the cloud. Measures that minimize downtime—for instance, redundancy, failover systems, and planned maintenance windows—keep critical IT functions accessible, minimizing the impact on operations. Tools and services that improve data consistency, provide fast data recovery, and offer disaster recovery services help mitigate data transfer risks, data integrity risks, and system availability risks.

Compatibility Issues

Addressing compatibility issues is a vital part of cloud migration. Challenges related to legacy systems, data formats, and application compatibility must be carefully managed. This can mean updating or replacing non-cloud–compatible systems, transforming data for cloud compatibility, and altering applications to prepare them for access and functioning in the cloud or using cloud-based services to achieve this goal.

Legacy Systems vs. Modern Cloud Infrastructure

Legacy systems are based on legacy technology that is likely not directionally compatible with cloud-based technology. That may include anything from low-level dependencies on the operating system, hardware, all the way to old and out-of-support software or integration with a cloud service. Cloud computing is inherently scalable and distributed, but the architecture of legacy systems (which may be monolithic) can limit your ability to capitalize on the overlapping scope of cloud benefits.

Legacy systems are based on older technology that may not be fully compatible with cloud-based infrastructure. This incompatibility can stem from low-level dependencies on operating systems, hardware, outdated software, or integrations that were not designed for the cloud.

Cloud computing is inherently scalable and distributed, but legacy architectures—especially monolithic systems—can create limitations that make it difficult to fully leverage cloud benefits such as elasticity, automation, and cost efficiency.

The final structure for solutions includes replatforming, which involves updating the application's underlying platform without changing its core architecture; refactoring or rearchitecting, which involves modifying or redesigning the application to be cloud-native; and replacing, which involves replacing the legacy system with a cloud-native solution.

Data Format and Structure

Data residing in legacy systems might be in formats or structures not suitable for the cloud. Many cloud platforms and services will either suggest or mandate specific data formats to achieve optimal performance, accessibility, and scalability. Furthermore, the data models in legacy systems may not fit in the more flexible, schemaless data models found in cloud databases. Here's a broad categorization of the solutions in this space:

- **Data Transformation:** Preparing or converting data in formats that are compatible with cloud services, such as JSON or CSV, to make data ingestible by cloud-based databases and analytics services.
- **Data Modeling:** Adapting the data model to more closely align with cloud-native databases, like migrating from a structured relational model to a place where your data can be free (document-based) if it makes more sense for the needs of the application.
- **ETL Processes:** Employing extract, transform, load (ETL) tools to automate the conversion and migration of data, ensuring it matches the target cloud environment's specifications.

Use Cases in Cloud Migration

Data transformation is an essential part of every data processing pipeline. In particular, data transformation is important to ensure compatibility with the new cloud environment. It includes the processes of preparing, converting, and refining data to fit cloud data storage formats, services, and the overall business goals of the migration project.

1. **Standardizing Data Formats:** During cloud migration, the data derived from various sources may need to undergo transformation into a uniform format compatible with cloud-based services and tools. This guarantees seamless integration and proper accessibility across cloud platforms.
2. **Cleansing Data:** The process of migration to the cloud can offer an opportunity to cleanse data by removing duplicates, rectifying errors, and filling in missing values. Cleaner data in the cloud improves efficiency—by storing only what you need in a cost-effective way—and accuracy through proper analysis.
3. **Transforming Schema:** Cloud databases and storage solutions might require data to adhere to specific schemas. This is why data needs to be transformed to conform to these schemas in order for applications and analytics tools to operate properly in the new environment.
4. **Optimizing for Cloud Analytics:** Data may need to be aggregated, enriched, or reformatted to support cloud-native analytics and business intelligence tools, enabling better decision-making and insights.
5. **Ensuring Compliance and Data Privacy:** Transforming data to comply with **regulations** (e.g., **GDPR, HIPAA**) is crucial when migrating **sensitive or personal data**. This may involve **anonymizing, encrypting, or restructuring** data to maintain **compliance** during migration.
6. **Reducing Data Volume:** Compression and deduplication techniques can help minimize the amount of data that is being migrated to the

cloud, thus enabling faster migration while lowering storage costs as well.

Tools and Their Application in Cloud Migration

Beyond just a step in the process, transforming data is an opportunity to optimize data to live in the cloud environment, ensuring data is clean, compliant, and structured to take full advantage of cloud computing. Data transformation tools and approaches should be chosen based on the objectives of the migration, like cloud service compliance, improved data quality, and advanced analytics functionalities.

1. AWS Glue, Google Cloud Dataflow, and Microsoft Azure Data Factory are cloud-native ETL services that offer integrated solutions for data transformation within their respective ecosystems. They provide seamless integration with other cloud services, making them ideal for cloud migration projects.

2. Apache Spark is a distributed processing system used for handling large-scale data processing across clusters of machines. Ideal for batch and real-time data processing, it is also ideal for bringing vast amounts of data to cloud.

3. Informatica PowerCenter and Talend are comprehensive data integration and transformation ETL tools. Informatica offers a solution for data integration for the enterprise grade across different sectors such as telecom, healthcare, finance, and government. Talend also follows a similar strategy with having separate software for dedicated functions like data preparation (Talend Data Preparation), data quality (Talend Data Quality), application integration (Talend Integration Suite), and big data management (Talend Big Data), respectively. Both are well suited for use cases where you want to merge data from multiple sources and perform complex transformations. Thus, when it comes to enterprise cloud migration at large scale, both are the right tools.

4. Pandas and dbt (Data Build Tool) are best suited for focused data manipulation and transformation tasks. Designed for data science and classical machine learning applications, Pandas is a fast, table-

oriented data manipulation system based on Python, supporting a wide range of processes from data preparation to data cleaning through its advanced data structure, DataFrame. dbt, an open-source ETL tool, enables data analysts and engineers to transform data in tables and views using SQL-based commands. These tools are particularly useful for granular data processing before loading it into cloud platforms or orchestrating it within cloud data warehouses.

5. Apache NiFi provides an extensible and configurable mechanism for on-demand data streaming as well as batch processing on cloud migration, facilitating additional layers of real-time transformation of data.
6. Trifacta is a data-wrangling and preparation solution that provides an intuitive interface for data cleaning/preparation for integration to cloud analytics cloud platforms. It helps analysts to discover, convert, and beautify raw data to clear and structured formats.

Application Compatibility

The difference in operating environment may cause compatibility problems of applications after moving to the cloud. This may include anything from specific library or software version dependencies, assumptions on the underlying hardware, or differences in architecture. Moreover, the dynamic nature of cloud environments could be trouble for the applications that are tightly coupled or tightly integrated or hard-coded. Here are a few common approaches to resolve compatibility:

- **Containerization:** Encapsulating applications along with required dependencies inside containers to maintain uniformity among various platforms, such as cloud-based systems.
- **Managed Services:** Using managed services provided by the cloud provider (for example, managed database, container orchestration services), so that differences in underlying infrastructure can be abstracted away.
- **Application Modernization:** Updating application code to adhere to 12-factor app principles for cloud-native applications, which include

using environment variables for configuration, building stateless processes, and managing dependencies explicitly.

Addressing compatibility issues during cloud migration is not a one-size-fits-all process; it requires a thorough assessment of the existing systems, a clear understanding of the target cloud environment’s capabilities and limitations, and a strategic approach to modernization that aligns with the organization’s goals.

Cloud Migration Security

The security of cloud migration refers to the methods and practices required for maintaining data, applications, and so on, during their shift from on-premises infrastructure to a decentralized configuration, which is also known as *public or private cloud hosting*, depending on the requirements. This pipeline needs more than just planning before the migration and security post-migration. There are greater risks with cloud infrastructure than on-prem, and any migration from on-prem to the cloud brings more risks because of this. However, both problems can be mitigated with more secure infrastructure and practices. As such, security in migrating to the cloud is not merely about mitigating the risks but also about harnessing all potential business benefits the cloud can confer for augmenting better cyber-resilience.

What Makes Migration to the Cloud Such a Necessity?

When your organization is moving to the cloud, it can be easy to overlook the complexities involved, but having a detailed breakdown of cloud migration in terms of the different types and their relative workloads will allow you to make the process straightforward and seamless every time. A lot of organizations migrate to the cloud to gain agility, accelerate time to market, and save costs. To realize these advantages, an organization must methodically plan and orchestrate the transition of data and applications, while moving from on-prem servers to the cloud. Ultimately, a successful migration plan will specify which applications and data will be migrated, include a timeline detailing when the migration occurs, and identify who is responsible for each phase of the process. By addressing these areas, organizations can concentrate on a seamless transition while preparing for what cloud computing can provide.

What Is Cloud Migration Security?

Cloud migration security entails a set of procedures and protocols needed to secure data, applications, and workloads during their migration from on-premises infrastructure to the cloud. Migration is a complicated affair and requires meticulous planning, assessment of any potential risks, and must be secured through appropriate measures to mitigate the risks.

The first step in cloud migration security is determining which workloads will be migrated and identifying the associated risks. This step may involve separating files or applications that need more support with security. Sensitive data, for instance, may require encryption prior to migration, whereas mission-critical applications may require additional layers of protection.

Once workloads are identified, the next challenge is to determine how these workloads will be migrated. This step usually includes transmitting them over secure channels such as encrypted proxies, VPNs, or SSL encryption to secure the data while in motion. After data and applications are migrated to the cloud, protecting them becomes just as important. When migrating to the cloud, organizations must go through four key phases to ensure a secure transition:

1. **Assessment**—This phase is about gathering data about your current on-premises infrastructure to figure out which workloads are ideal for migration. This will assist you in the next step of developing a migration plan.
2. **Planning**—This phase is almost all about creating a plan for your new cloud infrastructure. It requires mapping out the steps to migrate the workloads and all the data. At this stage, organizations will also choose what cloud service provider would be most suitable

for their business, giving the nature of data and what they intend to do with it.

3. **Execution**—In this phase, you carry out the migration. You may opt to move the data manually or use migration tools to move the data and set the configurations for the new cloud environment. In addition, testing your applications to ensure they are running properly in the new environment is essential.
4. **Post-Migration**—After the migration is complete, the organization must have a plan for monitoring performance. You must ensure that the data is backed up correctly. Also, you need to be on the lookout for any problem that may arise or that may pose a risk to your environment.

Risks of Cloud Migration

While implementing a cloud migration, an organization should account for certain risks prior to executing the migration. One of the biggest risks is data loss. Cloud data is stored on servers that are owned and maintained by the cloud provider. If these servers go down, the data is gone. Hence, no matter what reason you store data in cloud, it is wise to always have a backup.

While security is an advantage of going to the cloud, it’s also one of its biggest pain points. In the cloud, your data is potentially accessible to anyone with an Internet connection. This means that hackers or other malicious actors can access your data, provided they get around authentication and authorization measures that are in place. So, to avoid taking this risk, businesses have to maintain the encryption of their data and also implement strong security.

Vendor lock-in is another concern. By storing your data in a single cloud provider, you are relying entirely on the CSP to keep your data safe and accessible. The issue comes in when you want to switch providers. Moving your data out of the current provider’s servers may be difficult. This could lead to higher costs and frustration down the line.

Using multicloud architectures is an option to decrease reliance on a single vendor, but if your organization does not have a proper multicloud operational model and strategy, it may add additional complexity to managing the resources across different cloud ecosystems.

Preparing for Cloud Migration Security Concerns

Next, let's look at some of the critical security concerns that need to be taken care of during and even after migration to protect the data.

API Vulnerabilities

Cloud application programming interfaces create a bridge between cloud applications, data, and infrastructure, and they can become a major vulnerability in cloud data security. APIs can have poor authentication and authorization controls, no sandbox protection, and excess privileges. Addressing these vulnerabilities should be part of a comprehensive assessment that organizations must do when migrating data to the cloud.

Security Blind Spots

Cloud data may also be at risk due to security blind spots in the cloud infrastructure. Problems such as using SaaS applications for sensitive data and establishing shadow IT networks are common in some cloud environments. Organizations must be cognizant of these potential vulnerabilities when migrating to the cloud and take action to mitigate them.

Compliance Requirements

Many organizations must adhere to regulatory requirements when transferring data to the cloud. Compliance with security requirements can pose a significant challenge for organizations, particularly if the cloud provider does not meet these requirements.

Data Loss

Finally, the cloud migration process can increase the risk of data loss. This is especially true if the cloud provider lacks sufficient controls to protect and recover the data in the event of a security incident. Businesses can take several steps to help prevent data loss during cloud migrations. Strong encryption and authentication tools must be enforced for data in motion, while digital access to sensitive data should be restricted and regularly audited during the migration process. It’s crucial to maintain backups of important data in a system separate from the migration plan. Organizations should implement a phased migration strategy that allows for incremental and managed transitions. Following best-practice security measures is essential, including proper decommissioning through the removal and sanitization of all devices, drives, and servers from the source system. Finally, working with a CSP that maintains robust security controls and guidelines is vital for protecting data throughout the migration process.

Effective Safeguarding for Executing a Cloud Migration

Although cloud migration introduces potential security risks, organizations are able to take viable steps to secure their applications and data while taking the leap toward the cloud. The transition should always be in compliance with the necessary regulations, follow vulnerability addressing, and preserve the integrity of data—an important aspect from a security viewpoint to make it a secure transition with minimal business risks. Security alone is not the road; cloud migration security requires careful planning, risk assessment, and implementation of best practices. The following sections describe important safeguards for achieving integrity, compliance, and security during the entire lifecycle of a migration. Feel free to adapt those best practices to your organization.

Understanding Your Data and Compliance Requirements

Organizations getting ready to migrate to the cloud need to know exactly how their data is classified, how it is used, and what its compliance requirements are. The migration teams must also assess existing data storage policies, how long they need to retain the data, and how to access the data after migration. The first step to securing cloud data is to understand its contents and how it will be used or disposed of.

Organizations must also assess their compliance obligations—namely, GDPR, PCI DSS, and HIPAA—which make explicit requirements with regards to systems that process sensitive personally identifiable information (PII). Choosing an appropriate cloud service provider is just as important because the provider should meet constituent security industry standards, apply sound security controls, and support relevant regulatory frameworks.

Safeguarding APIs and Access Controls

APIs control access between environments, cloud services, applications, and infrastructure, all which are high-value target for security vulnerabilities. To mitigate the risks, companies must ensure that their APIs follow very strict practices in authenticating and authorizing users. To protect against unauthorized consumers, OAuth, API gateways, and access tokens can be employed, while TLS and other communication-encryption methods ensure secure communication.

Adopting the principle of least privilege (PoLP) ensures that you are distributing only appropriate permissions, thus reducing API exposure. Additionally, organizations need to constantly review API traffic for irregularities and protect against exploitation by malicious **Actors**. Implementing **a comprehensive API security strategy** strengthens cloud defenses and reduces the risk of **data breaches**.

Encrypting Your Data During Transit

Transferring data during cloud migrations may introduce additional security vulnerabilities, and a basic practice of end-to-end encryption is a great way to protect sensitive information. It usually includes an encryption protocol like Transport Layer Security that provides security by encrypting all data prior to sending it out from the source system and decrypting it on arrival at the target system. There are a number of different encryption algorithms to choose from, depending on how much protection you require, but the vast majority use contemporary industry standards such as AES-256 encryption or equivalent.

Another tip is that companies need to ensure that any encryption keys and access credentials are stored securely and backups are done regularly so that there is no data loss. Working with a cloud provider offering incorporated encryption solutions can make this process more manageable. That said, companies should still exercise due diligence to qualify themselves for having the right tools and security measures in place prior to starting the migration.

Limiting Data Access During Cloud Migration

While migrating the heart of an enterprise—its data—to the cloud, limiting the access period to the data is the first critical step for any organization, big or small, toward securely migrating to the cloud. Access control during migration requires defining and enforcing authentication and authorization rules at the user level, implementing strong two-factor authentication (2FA) procedures, and leveraging the cloud provider's native security policies. Data protection measures should include enabling encryption before transfer, regular access audits during migration, vulnerability scans on sensitive systems, and prompt removal of credentials for terminated employees.

Employing a Phased Migration Strategy and Risk Mitigation

To mitigate the risk of data loss and interruption of services in a cloud migration, organizations need a phased migration strategy. That means transferring workloads in small, manageable batches while verifying security at every stage. Before starting the migration, the organization should ensure that backed-up critical data has its corresponding restore mechanisms implemented in case migration fails. As part of a zero trust security model, organizations should validate continuously at every stage of migration to verify that only verified entities are able to access systems.

Implementing Decommissioning and Sanitization Activities

Organizations must properly decommission legacy systems to prevent residual data exposure. *Decommissioning* involves inspecting all your devices, drives, and servers that remain in your data center. Maintain a checklist that documents all of that hardware, so you can be sure to remove everything from your current cloud or on-premises storage servers.

Securely delete any data you have stored in off-site locations, too. Also, you might want to give your cloud infrastructure providers a security audit to verify whether they can protect their systems and monitor them in a real-time manner.

Following NIST data sanitization guidelines, systems should be thoroughly wiped and decommissioned, to guarantee that no retrievable data remains. The final step involves conducting a comprehensive security audit to verify that all residual data has been removed and confirm no security gaps remain.

Formulating a Security Plan

A security plan turns out to be the key component for ensuring solid cloud data protection while using resources, migrating to the cloud, or storing data, etc. By taking proactive approach, you can identify potential weaknesses and take steps to protect yourself against them.

Understanding your organization's security needs and migration process risk is crucial to drafting a security plan. Perform a risk assessment to determine potential threats and vulnerabilities across the three states of data —data in transit, archived data, and data in use. Then formulate protocols to tackle these identified threats, which could involve firewalls, encryption protocols, and subnetting. The team responsible for executing the migration process should also be clearly identified in the security plan. Finally, create contingency plans for potential security breaches.

Maintaining Data Protection and Integrity

Data protection and integrity have a direct impact on an organization's most critical asset: information. Compliance with various regulatory standards, such as GDPR and HIPAA, is also vital. Noncompliance penalties can be severe. However, maintaining data privacy and integrity is not only about avoiding fines and public opinion backlash; reliable data is necessary for making impactful decisions.

Ensuring that coding and other key documents are properly version controlled alongside using full encryption on data at rest, in transit, and in use will help maintain a secure environment. Alongside automated data backup solutions and data loss prevention tools to track and stop breaches, regular vulnerability scans and penetration tests should also be undertaken. Running these tests ensures redundancy and prevents accidental loss or corruption of data. Breaches should be detected and prevented before they occur, so you should have continuous security monitoring and threat detection mechanisms in place.

Confirming Security Measures

Verifying security implementations is an important post-migration task. To validate the effectiveness of security controls, organizations should conduct independent security audits.

A red team exercise, which simulates real-world cyberattacks, is used to assess the resilience of cloud defenses. Real-time surveillance by Security Incident and Event Management (SIEM) systems is critical for detecting anomalies and identifying potential threats.

A regular review of access controls is necessary to remove permissions from users who no longer need them, ensuring least-privilege access is maintained.

Last but not least, disaster recovery procedures should be created and tested to allow systems to be quickly restored in case of failure. An organization is unable to ascertain the safety of its data or the efficiency of the implemented safeguards without adequate validation.

Cloud Migration: Security Checklist

A security checklist can assist in covering each and every cloud migration security aspect at all levels and stages. Organizations should follow a structured approach to expand their security across the cloud from pre-migration risk assessments to post-migration security maintenance.

Pre-Migration Security Preparations

Organizations looking to migrate to the cloud should outline a security framework that defines migration objectives and identifies migration risks, lessons learned, costs, and benefits, as well as compliance with security policies, before initiating cloud migration. This phase lays the groundwork for a safe transition:

- **Risk Assessment and Planning**
  - Carry out risk assessments and then assess possible weaknesses in the existing infrastructure.
  - Analyze threats that can occur while migrating and prepare contingency plans.
  - Set clear migration goals, considering security implications, and evaluate migration scope and potential vulnerabilities.
- **Identity and Access Management (IAM) and API Security**
  - Change the default passwords on servers and other critical devices. Don’t use common or repetitive patterns.
  - Configure access based on the principle of least privilege.
  - Consider federated authentication to allow access with multiple credentials and reduce credentials sprawl.
  - Implement strong passwords and two-factor authentication policies.
  - Set up user roles and permissions for access control.
  - Use cloud-based identity management solutions like Azure Active Directory.
  - Set up user activity monitoring for suspicious logins.
  - Automate the access lifecycle, such as decommissioning credentials when an employee exits.

  - Apply maximum zero trust security levels during migrations and configurations.
  - Establish API security policies to limit third-party platform access.
- **Cloud Service Provider (CSP) Evaluation**
  - Carefully examine what security protocols and measures the CSP utilizes.
  - Verify CSP compliance with risk and governance frameworks such as GDPR, SOC 2, HIPAA, and other industry standards.
  - Check the record of the service provider for safe operation and study the service contract.
  - Evaluate the CSP infrastructure, such as firewalls, email encryption, and system maintenance.
  - Check the backup and security protocols of the CSP for data recovery in emergencies.
  - Review the geographical placement of servers for compliance and speed considerations.
  - Assess the provider’s incident response plan for handling security breaches.
  - View the service-level agreements (SLAs) that a CSP provides for security assurances.
  - Check and change the CSP’s firewall and network segmentation strategies if needed.
  - Ensure the CSP provides regular security patching and updates.
  - Make sure that data residency policies conform to jurisdictional laws.
- **Data Preparation and Security Measures**
  - Conduct data cleanup for fewer redundancies, smaller migration size, and greater speed.

  - Determine how to treat data associated with ex-employees.
  - Perform a comprehensive data backup to a secondary secure storage system.
  - Implement data masking techniques to protect sensitive information.
  - Secure your APIs and encrypt data before transferring.
  - Evaluate the security of any third-party vendors with which the migration is associated.
  - Plan and configure the architecture for the new environment.
- **Network and Firewall Security Measures**
  - Segment the network (including cloud) securely.
  - Implement a whitelist approach toward the firewall settings, only permitting the IP addresses that are known.
  - Set alerts for log-in attempts beyond normal business hours or locations.
  - Ensure access control policies are routinely inspected for protection of authentication and authorization.
  - Employ antimalware such as an antivirus or web application firewalls (WAFs).

Security During Migration

This phase ensures that data, applications, and infrastructure are securely transitioned during migration while minimizing security risks:

- **Data Transfer Security**
  - Send data over encrypted communication (such as VPN, TLS).
  - Monitor access logs for unusual activity during migration.
  - Apply RBAC to restrict migration privileges.

  - Ensure that all user accounts have MFA/2FA and strong authentication enabled.

  - For movement of data, use secure file transfer protocols (such as SFTP, HTTPS).

- **Continuous Monitoring and Threat Detection**

  - Perform penetration tests and vulnerability scans of cloud environments.

  - Monitor SIEM logs for signs of anomalies or unauthorized access attempts.

  - Implement real-time security monitoring to detect abnormal activities.

  - Assign a team to review logs for unauthorized access attempt or suspicious activities.

- **Cloud Environment Security Configurations**

  - Implement zero trust security models as much as possible during the migration.

  - Validate the safety and integrity of the encryption keys and API tokens before use.

  - Do heavy security testing before moving workloads into production.

  - Keep all security configuration updated and patched from time to time.

Post-Migration Security Maintenance

After migration, organizations still need to proactively protect their new environment against new incoming threats while confirming the integrity of the migrated systems. This means organizations need to keep on tracking their cloud environment, perform security checks, and take preemptive steps to find and fight possible threats:

- **Post-Migration Security Validation**

  - Perform an exhaustive security audit to establish the integrity of the migration environment.
  - Ensure that everything removed from service has been wiped.
  - Implement separation of duties to address internal risks.
  - Review network firewall configurations to prevent unauthorized access.
  - Implement continuous monitoring solutions (such as SIEM, IDS/IPS).
  - Periodically rotate encryption keys and API tokens to avoid their use without authorization.
  - Conduct compliance checks to ensure the business remains compliant with industry regulations.
  - Test the disaster recovery plan (DRP) to ensure a fast return to a normal operating state in the case of failure.
- **Continuous Monitoring and Security Audits**
  - Enable real-time security monitoring to identify abnormal activities.
  - Exercise periodic red team or cyber attack simulation exercises.
  - Audit IAM policies and disable inactive or unauthorized accounts.
  - Regularly audit the cloud environment and the CSP’s performance.
  - Assign a team to review logs for unauthorized access attempts or suspicious activities.
  - Establish a continuous enforcement process around best practices and protocols for security.
- **Threat Detection and Response**
  - Set up IDS for anomaly detection.
  - Monitor cloud logs for failed authentications and strange IP traffic.

    - Create automated alerts for potential security incidents.
    - Perform an initial vulnerability scan after migration.
- **Regulatory Compliance and Data Protection**
    - Periodically revise your security policies to keep pace with any regulatory changes.
    - Ensure continual GDPR, HIPAA, and PCI-DSS compliance for workloads with sensitive data.
    - Keep data protection impact assessments (DPIAs) up-to-date.

Quality Engineering: The Heart of Cloud Migration

As the cloud takes on an increasingly important role in the enterprise digital transformation story—where speed and agility are the holy grail of IT strategy—*quality engineering (QE)* becomes mission-critical in enabling your organization to move to the cloud faster, better, and more securely. Based on perspectives from various industry leaders, the following sections highlight the necessity of QE in solving challenges related to cloud migration.

Enterprises in the process of cloud adoption must ensure a laser-focus on quality to achieve speed-to-market and business performance goals. This requires holistic quality engineering, which not only defines quality culture but also drives quality across the entire software development lifecycle. QE emphasizes maximizing automation, using analytics and artificial intelligence to improve performance, lower costs, and accelerate time to market.

The Landscape of Cloud Migration Challenges

Challenges associated with the cloud transition process include the creation and development of cloud environments, obstacles to moving workloads, and optimization gaps. These challenges often stem from not having a clear strategy linked to business objectives, cloud sprawl, excessive costs, or even nonoperation of a service due to a high-impact vulnerability with no time to fix it or due to a lack of skills needed to maintain the service. These issues highlight the importance of a clear, quality-focused strategy to navigate the transition successfully.

Quality Engineering’s Strategic Blueprint

The quality engineering approach to cloud migration addresses the entire software delivery lifecycle to ensure business as usual even as quality assurance testing continuously changes. It aims to remove scalability problems, server crash, database errors, and so on, which are speed breakers. The QE test strategy is divided into three phases with corresponding activities:

1. **During Advisory Phase (Pre-Migration):** Quality engineering facilitates test case coverage calculation, maps automation test suites to ensure maximum coverage, and develops sanity and regression suites while determining the migration sequence of components. Baseline performance tests are scripted and established as benchmarks for later comparison, and test suites are executed to validate performance. Apart from this, this stage also includes setting up key performance indicators (KPIs) and ensuring data classification and risk assessment mechanisms. These steps take into account the early phases of the migration; the purpose is to address sensitive data handling and potential vulnerabilities, building a strong foundation for a safe and efficient migration.

2. **Assurance Stage (During Migration):** In the assurance stage, QE begins testing on the most recent build within a proof of concept (POC) environment, ensuring that the migration aligns with performance expectations. Higher cloud tiers are tested for scalability, and continuous integration tests are run to ensure the workflow remains undamaged. This stage also includes testing defects in an iterative manner and fixing issues quickly to reduce risk. A sign-off is carried out on every iteration of QA builds, facilitating a controlled and smooth migration process. QE integrates user acceptance testing and monitoring tools to proactively

overcome operational challenges to ensure optimal migration in real time.

3. **Validation Stage (Post-Migration):** Validation is performed when the process is complete to ensure a successful post-migration experience. Then a release candidate build is generated and run through sanity and smoke tests to ensure system stability. Comparative performance metrics are analyzed to assess the impact of migration and determine whether the operation was successful or if any performance issues emerged. The standard method for system validation includes User Acceptance Testing (UAT) and continuous system monitoring to observe performance trends over an extended period. When you add all of these features, you have QE that assures absolute data security and high operational reliability, along with complete functioning in all systems post-migration.

Ensuring Data Security and Navigating Cloud Migration with Precision

Given that many organizations experience security incidents in their cloud infrastructure, ensuring data security during cloud migration is critical. In this landscape, it is imperative to identify and assess potential security threats through a quality engineering approach. Organizations can then proactively mitigate these risks by integrating continuous testing, monitoring, and reviews into a comprehensive cloud migration strategy. While this is an effective approach for performance, scalability, and security, it also emphasizes data classification, risk assessment, encryption, access control, monitoring, auditing, and compliance. These measures minimize data leakage and prevent sensitive data from being under the provider’s control.

Quality engineering ensures that moving to the cloud makes sense, not just technically, but as a way to deliver more agility and lower costs, as well as gain a competitive advantage. Organizations can ensure that best-quality data practices are adopted while embedding strong data security practices within the migration process. With the expertise of quality engineering teams, enterprises can approach the intricacies of cloud migration, achieve seamless migration and operational efficiency, and unlock business objectives in the modern economy.

Network and Connectivity Considerations

Real-time threat detection and monitoring of network activity require continuous monitoring and assessment. Restricting access and segmenting means that controlling permissions, restricting access, and implementing segmentation are necessary to address these challenges. Encryption protects data in transit and in the cloud, whereas compliance with security policies and standards means that regulations are followed in the cloud.

The Maintenance of Network Security Postures During and After Migration

A network security posture is the status of your network hardware and software, policies and processes, and protections. It ensures your network protects against hackers, penetration, and data leakages. Maintaining or improving your network security posture during and after a cloud migration requires the following:

- **Continuous Monitoring and Assessment:** You need to implement tools and processes to continuously monitor network traffic and activities, identifying potential security threats in real time.

- **Access Controls and Segmentation:** You can use network segmentation to divide the network into smaller, manageable segments, each with its own set of access controls, to limit the spread of potential attacks and manage permissions more effectively.

- **Encryption:** You must ensure that data in transit and at rest is encrypted, securing the data from interception during the migration and when stored in the cloud.

- **Compliance with Security Policies and Standards:** You must ensure compliance to industry standards as well as regulatory requirements for network security while modifying policies and controls in the cloud environment per the compliance requirements.

Continuous Monitoring and Assessment

Cloud environments require continuous monitoring and assessment during and after migration to ensure network security, performance, and compliance. Various tools are available: Some are cloud-native tools offered by the provider of the cloud service, some are tools from traditional IT vendors, and some are open-source projects. Here are some examples:

Cloud Provider Tools:

- AWS
  - Amazon CloudWatch: Provides monitoring and observability of AWS resources and applications, offering logs, metrics, and event data. It is recommended for tracking application performance and system health in real time, making it ideal for organizations running dynamic workloads on AWS.
  - AWS CloudTrail: Enables governance, compliance, operational auditing, and risk auditing of your AWS account by logging and monitoring account activity. It is best suited for maintaining compliance in regulated industries, because it tracks all user activity and API calls for audit readiness.
  - Amazon GuardDuty: Offers threat detection that continuously monitors for malicious activity and unauthorized behavior. It is ideal for detecting anomalies in network traffic or unauthorized attempts to access resources, particularly for securing hybrid cloud workloads.
- Azure
  - Azure Monitor: Collects, analyzes, and acts on telemetry data from Azure and on-premises environments. It is useful for organizations needing centralized telemetry from hybrid environments, supporting proactive issue resolution and performance optimization.

    - Azure Security Center: Strengthens the security posture of data centers, providing advanced threat protection across hybrid workloads in the cloud and on-premises. It is recommended for detecting vulnerabilities and enforcing security policies in hybrid IT setups.
    - Azure Sentinel: Provides a scalable, cloud-native, security information event management (SIEM) and security orchestration, automation, and response (SOAR) solution. It is best suited for organizations needing advanced threat detection and automated incident response across cloud and on-premises infrastructures.
- Google Cloud Platform (GCP)
    - Google Cloud Operations Suite (formerly Stackdriver): Offers monitoring, logging, and diagnostics to ensure visibility into the health, performance, and availability of cloud-powered applications. It is ideal for managing and troubleshooting multicloud environments, with deep integration into GCP services.
    - Google Cloud Security Command Center: Provides risk and threat identification to Google Cloud assets across GCP services. It is recommended for identifying security risks and policy violations in GCP deployments, enabling proactive mitigation.

Traditional IT Vendor Tools:

- Cisco
    - Cisco Secure Cloud Analytics (formerly Stealthwatch Cloud): Monitors network traffic and user behavior, detecting anomalies and threats in real time. It offers visibility and security analytics across your cloud and on-premises environment, utilizing behavioral modeling for threat detection. It is best for detecting insider threats or abnormal behavior in hybrid cloud networks, offering advanced anomaly detection.
    - End-to-End Visibility with Cisco ThousandEyes: Provides crucial end-to-end visibility, facilitating successful cloud migration and digital experience monitoring. It is ideal for organizations

undergoing complex migrations, enabling proactive issue identification and ensuring uninterrupted service delivery.

Open-Source Options:

- Nagios: An open-source software tool that offers monitoring and alerting services for servers, switches, applications, and services. It alerts users when things go wrong and alerts them again when the issues are resolved. It is best for small- to medium-sized enterprises needing basic monitoring of their on-premises and cloud infrastructure.
- Prometheus: A powerful open-source monitoring and alerting toolkit initially built by SoundCloud, now part of the Cloud Native Computing Foundation (CNCF). It's particularly well suited for monitoring cloud-native environments. It is recommended for cloud-native application monitoring, particularly in Kubernetes-based ecosystems, due to its efficient metric collection and alerting capabilities, but can be utilized if Prometheus/OpenTelemetry are used as organizational frameworks for metrics, and if support for this metric format exists within the monitored tools.
- Elastic Stack (formerly ELK Stack): Combines Elasticsearch, Logstash, and Kibana to provide powerful logging, monitoring, and visual analytics capabilities. It's widely used for centralizing and analyzing logs and metrics from various sources. It is ideal for organizations needing log aggregation and analytics, especially those managing hybrid environments with diverse data sources.
- Grafana: An open-source platform for monitoring and observability, compatible with multiple data sources like Prometheus and Elasticsearch, offering powerful visualization tools for metrics and logs. It is best for teams focused on custom dashboards and multidata source integration, providing real-time observability across IT environments.

Each of these tools and platforms has its strengths and particular use cases, and many organizations will find that a combination of several tools best meets their monitoring and assessment needs. The choice depends on the specific requirements, such as the complexity of the cloud environment, the level of integration needed with existing tools, compliance requirements, and budget constraints.

Access Controls and Segmentation

Access controls and segmentation are critical components of a comprehensive network security strategy, especially as organizations transition to cloud environments or manage complex hybrid networks. These concepts are essential in minimizing the potential impact of security breaches and in ensuring that only authorized users and systems can access sensitive resources.

Network Segmentation

Network segmentation involves dividing a larger network into smaller, discrete segments or subnets. This division is typically based on factors like function, security level, or department. Each segment can have its own unique set of policies and controls, tailored to the specific needs and security requirements of the data or applications it contains.

Benefits of network segmentation include

- **Enhanced Security:** By isolating segments of the network, you can contain security breaches within a limited area, significantly reducing the potential impact of an attack. This containment strategy is akin to compartmentalizing a ship to prevent it from sinking if part of it is breached.

- **Improved Performance:** Segmentation can reduce network congestion by limiting broadcast traffic to within smaller network segments, thereby improving overall network performance.

- **Simplified Compliance:** For organizations subject to regulatory compliance, segmentation can help by isolating the systems that process sensitive information, making it easier to apply specific controls and conduct audits.

Access Controls

Access controls govern which users can see or access resources within a computing environment, and there is a fair bit of variance between the application of access control in cloud environments, on-prem systems, and the transport networks that connect them. On-premises access might be controlled by something like Active Directory (AD) or Cisco ISE, whereas access to cloud-based resources might be governed by a cloud IAM framework such as AWS IAM or Azure AD. Likewise in transport networks, transport access controls are implemented in firewalls, VPN policies, or SD-WAN configurations to secure connectivity as migration occurs. These are bilevel data, application, and network pathways through which anything will be obtained on each and every conceivable layer of access to information.

Types of access controls:

- Access is granted according to the user role in the organization. For on-premises, roles can exist within Active Directory, and in the cloud, IAM roles are assigned specific permissions to cloud resources such as storage or compute instances. In transport networks, roles can decide who can manage or access routers, VPNs, or connectivity configurations.
- Access decisions are per user attributes (location, time, device type, etc.). An example of this could be that ABAC will only allow access to sensitive cloud resources if the user connects over a defined range of IP address. ABAC might tighten access to on-premises environments during nonworking hours, or from a personal devices.
- Policies are strictly enforced by the system, making it ideal for high-security environments. MAC is a system-enforced method of restricting access to objects based on the sensitivity of the object and the clearance of the user. Unlike discretionary access controls, users cannot override these policies. This makes MAC particularly valuable in environments such as government or financial institutions, where strict compliance and non-negotiable security protocols are required. For example, in a cloud deployment, MAC

can be used to ensure that access to sensitive databases is only granted to users with the appropriate security clearance.

Implementing effective access controls:

- **Least-Privilege Principle:** You should grant only the minimum access required for tasks. In the cloud, this might involve restricting API-level permissions for developers. On-premises, users may be limited to certain shared drives. For network transport, access could be restricted to only the IPs necessary for secure data transfer.

- **Multifactor Authentication (MFA):** You must require two or more authentication factors to bolster security. MFA is critical for cloud console access, VPN connections, and administrative tasks on-premises, reducing risks from compromised credentials.

- **Regular Reviews and Audits:** You need to regularly audit access rights across environments. In a hybrid scenario, this may pertain to ensuring alignment between IAM roles for cloud resources, AD groups for on-prem systems, as well as firewall policies aimed at transport networks vis-a-vis the latest security policies.

Integrating network segmentation with robust access controls creates a multilayered security architecture that significantly enhances an organization's defense against internal and external threats. This approach not only limits the lateral movement of attackers within the network but also ensures that sensitive information and critical systems are accessible only to those who legitimately need access.

Segmentation plays a vital role across multiple network domains:

- **Data Center Network Segmentation**: In traditional and hybrid data center environments, segmentation involves creating isolated zones using VLANs, firewalls, or microsegmentation tools like VMware NSX. These zones restrict traffic between segments, reducing the attack surface and protecting critical workloads.

- **Cloud Network Segmentation**: Cloud-native tools, such as security groups, virtual network peering, and network ACLs provided by AWS, Azure, or GCP, enable segmentation at the virtual network level. For example, workloads in different subnets or regions can be isolated to prevent unauthorized communication.

- **Transport Network Segmentation**: When on-premises infrastructure is connected to the cloud, segmentation over transport networks ensures secure and optimized communication. Technologies like VPNs, MPLS, and SD-WAN can establish segmented tunnels, ensuring that only authorized traffic flows between on-prem and cloud environments.

For cloud environments, adopting a zero trust security model—where trust is never assumed and verification is required from everyone trying to access resources—complements segmentation strategies. Combined with strict access controls, such as identity-based access using IAM or role-based access controls, this approach strengthens overall network security.

Integrating segmentation across these layers and enforcing robust access controls helps organizations to effectively secure their network environments and reduce the risk of lateral movement by attackers. These strategies are indispensable in a comprehensive cybersecurity playbook, offering both proactive and reactive benefits.

Network Considerations for Cloud Migration

In the dynamic journey of migrating into the cloud, organizations are faced with a spectrum of considerations that ensure their network functions transition seamlessly, optimizing cost, performance, and security in the cloud environment. This migration, while offering a plethora of benefits, demands a detailed strategy that addresses pre- and post-migration networking considerations to harness the full potential of cloud computing.

Measuring and Ensuring Network Performance

Pre-migration considerations involve assessing cloud compatibility, costs, security, and design, while post-migration considerations include managing IP addresses, DNS changes, cost management, and ensuring high availability. Continuous performance monitoring is essential for maintaining optimal network functioning.

- **Pre-Migration Networking Considerations:** Assessing cloud compatibility, costs, security, and design compatibility.
- **Post-Migration Networking Considerations:** IP address management, DNS changes, cost management, high availability strategies, and continuous performance monitoring.

Consistently Measuring Network Performance

Effective network performance management hinges on consistency. Regular application of measurements helps IT teams spot trends, identify potential problems early, and rectify them before they become significant service disruptions. By using high-performance servers or dedicated workstations for testing and a variety of tools, organizations can gain a detailed understanding of their network’s performance, thereby making informed decisions to optimize it.

In essence, maintaining optimal network performance requires a proactive approach, which includes using reliable metrics and sophisticated tools. This strategy allows IT teams to ensure their networks are robust, efficient, and can support the organization’s digital activities without interruptions.

Enhancing network performance measurement strategies with tools, such as the TRex Traffic Generator, deepens testing capabilities by enabling more realistic and challenging network conditions to be simulated and analyzed. Additionally, using the specialized monitoring and diagnostic tools provided by hyperscalers like AWS, Azure, and GCP ensures that organizations can maintain optimal network performance in the cloud, benefit from integrated insights, and promptly resolve any issues to effectively support their cloud strategies.

Pre-Migration Networking Considerations:

- **Workload Evaluation:** Prior to migration, it’s crucial for organizations to evaluate which workloads are best suited for the cloud, taking into account factors such as compatibility with cloud environments and potential cost savings or implications. For example, high-bandwidth applications like video processing may require special consideration to avoid performance degradation or unexpected costs. Migrating workloads without this assessment could lead to unforeseen expenses, emphasizing the need for thorough planning and analysis. Extra care should be taken regarding network traffic requirements within the application's components or cloud environments, VPCs and regions, because networking output could be a big cost factor that is often overlooked.

- **Network Performance and Design Compatibility:** Understanding the performance requirements of your applications and how they will interact with the cloud's network architecture is paramount. This includes evaluating whether your current network topology supports low-latency, high-throughput connections to cloud services. For instance, some applications may rely on low-latency connections that require direct peering or the use of a high-speed dedicated link like AWS Direct Connect or Azure ExpressRoute. Additionally, applications with significant interdependencies may need network adjustments such as VLANs or SD-WAN configurations to optimize communication between cloud and on-premises environments.
- **Security and Compliance:** With workloads shifting to the cloud, the cloud network security posture is more difficult to manage and yet much more essential. This includes a set of security policies, tools, and practices that help preserve the security, confidentiality, and integrity of data. Organizations need to create policies and procedures to enforce security policies, like least-privilege access controls, or install firewalls and use network vulnerability management tools to continuously discover and remediate risks, as an example. **Incident response capabilities** must be established to **quickly manage breaches**, while **intrusion detection and prevention systems (IDS/IPS)** protect against **network attacks**.

In addition, there are tools such as AWS GuardDuty or Azure Sentinel that can be implemented for continuous monitoring and network assessment, which also helps ensure compliance with industry standards and regulatory requirements, and helps protect data and applications throughout the lifecycle of the migration.

Post-Migration Networking Considerations:

- **Verifying IP Address Management and DNS Changes:** When an organization is migrating to the cloud, IP addressing schemes and DNS changes need to be managed to keep the applications working and performing effectively. Planning for IP address management (IPAM) and DNS changes will mitigate downtime and connectivity issues post-transition.
- **Monitoring and Managing Cloud Costs:** Once you migrate to the cloud, monitor costs associated with networking services. Cloud providers or third-party solutions offer tools for organizations to optimize their cloud spending by detecting deviations from their resource use pattern, such as idle resources, suboptimal pricing model, or scaling not being done at the right place or at the right time.
- **Ensuring High Availability and Disaster Recovery:** A cloud environment provides invaluable opportunities to improve the availability and resiliency of applications, but this is not possible unless the network is thought of as being fundamental to achieving these goals. For example, network performance issues such as high latency or limited bandwidth can become bottlenecks during disaster recovery failover or reduce application uptime.

  To mitigate these risks, organizations must configure their cloud networking components, such as load balancers, redundant VPNs, or Direct Connect links, to support seamless failover and recovery across multiple regions. Services like auto-scaling and multiregion deployments should be paired with network redundancy to ensure traffic is dynamically rerouted during outages or peak usage. By addressing these network-specific factors, businesses can align their high availability and disaster recovery strategies with the demands of modern cloud architectures.
- **Continuous Performance Monitoring:** Continuous monitoring of network performance is crucial to ensuring that applications in the cloud meet user expectations. Application performance management (APM) tools, such as AppDynamics, and network performance monitoring solutions are indispensable in identifying potential

bottlenecks in network communication, such as delays caused by overloaded links or misconfigured routing.

For example, APM tools may be able to spot latency spikes or packet loss between application components hosted in separate cloud regions so that IT teams can optimize routing paths, increase bandwidth, or scale resources to meet demand. These insights help guarantee that high availability setups like load balancers and failover methods work seamlessly, without adding any further latency or performance degradation.

- **Adapting to Cloud Networking Models:** Understanding and adapting to the cloud provider's networking models—such as VPCs and concepts like subnets, security groups, and cloud routing—help ensure that your network architecture supports the scalability, security, and efficiency that cloud environments offer.

Measuring and Assessing Network Performance by Monitoring Key Metrics

IT administrators use metrics and specialized tools for comprehensive network performance analysis to maintain maximum performance and prevent possible bottlenecks and service quality interruptions. This all comes down to ensuring consistent measurement because network performance affects both migration success as well as its subsequent operations, which, in turn, affects the overall quality of service:

1. **Bandwidth:** Bandwidth is the amount of data rate that can be transferred over a network connection in a particular time period. This obvious yet important metric is used to estimate the possible bandwidth of a network link.

2. **Throughput:** Throughput is different from bandwidth because it determines the actual amount of data passed through the network. It can be influenced by several factors, including network congestion and hardware issues, so it should be monitored regularly.

3. **Latency:** Latency is the time needed for a data packet to find its way from point A to point B across the network. Lower latency is always

better, particularly for real-time applications and services.

4. **Jitter:** Jitter is the difference in latency on the same link over time. Voice over Internet Protocol (VoIP) and video streaming services are especially sensitive to timing inaccuracies, so inconsistency of latency can become problematic.
5. **Error Rates:** The rates refer to the percentage of corrupted bits over total bit transmission. High error rates may point toward faulty hardware, signal interference, or other network stability issues.

Network Performance Measurement Tools

Several sophisticated tools are available for IT professionals to measure these metrics accurately:

- **IxChariot (by Keysight Technologies):** This tool provides comprehensive network performance testing for applications, devices, and services. It can simulate a wide range of applications and measure key performance indicators like bandwidth, latency, and jitter.
- **iPerf:** This open-source tool is ideal for measuring throughput on IP networks. It supports tuning of various parameters and UDP data streams to test the performance under different conditions.
- **Spirent TestCenter:** This tool offers a wide array of testing capabilities for networks, devices, and services. It can simulate complex networking environments and measure performance across multiple metrics.
- **NetCPS:** This simple tool is focused on measuring the throughput between two endpoints. It’s useful for quick assessments of network capacity.
- **TRex Traffic Generator:** This open-source, high-performance traffic generator is based on the Data Plane Development Kit (DPDK). TRex has the capability to emulate realistic traffic from packet captures or by customizing traffic patterns, volumes, and flow characteristics. This capability is especially useful for stress-testing

networks to examine how they cope with different conditions, including high throughput and complex traffic scenarios. TRex's capabilities allow you to

- Generate L4-7 traffic based on custom profiles, providing an in-depth view of how your network detects and accepts specific types of data and applications.
- Measure throughput, latency, and packet loss using realistic traffic patterns, helping to identify potential performance bottlenecks, security vulnerabilities, or network inefficiencies and degradation.
- Enable advanced network security testing by simulating DDoS attacks or malicious traffic patterns, allowing for a comprehensive evaluation of network defenses.

Application Dependency Mapping (ADM) and (ADM) and Application Performance Management (APM): Navigating Secure Cloud Migration and Zero Trust Architecture

With the groundwork on security and performance monitoring laid, we can now delve into the roles of application dependency mapping (ADM) and application performance management (APM) in ensuring that applications migrate smoothly and continue to perform optimally in their new cloud environment. This discussion ties back to the importance of network segmentation and access controls mentioned earlier because ADM can help identify critical dependencies that must be secured.

Cloud migration is not just a technical issue but a strategic necessity, and if you don't have the right tools, it can go horribly wrong. Accountability and visibility, which are needed for successful cloud migrations, are presented by application dependency mapping and application performance management. These tools are essential to enable microsegmentation, also enforcing zero trust and ensuring that security and performance remain tightly coupled.

ADM: Charting the Cloud Migration Terrain

ADM provides the necessary insight to detect and comprehend the complex interdependency map of applications, databases, and services. Providing this insight is critical for secure cloud migration and designing zero trust networks with

- **Complete Visibility:** ADM tools reveal the links between applications and their dependencies on-premises or in the cloud. This level of mapping ensures that all sensitive components are migrated to the next infrastructure together to avoid performance issues due to missing dependencies.
- **Migration Plan with Impact Analysis:** ADM analyzes dependency so that the organization anticipates the consequences of migration that flow from the migration within the organization and design strategies that reduce the impact. It determines the workloads that need secure connectivity so that these can be prioritized for connection during the migration.
- **Microsegmentation Design:** ADM defines logical segments to restrict lateral movement, adheres to zero trust policies, and protects critical workloads. Sensitive data can be quarantined in assessed but protected secure enclaves where security is highly enforced but connectivity is still able to be maintained.

By turning the complexity into actionable intelligence, ADM makes cloud migrations secure, scalable, and efficient across hybrid and multicloud environments.

APM: The Compass for Cloud Optimization

APM enables applications to be both performant and secure in the migration process and beyond. APM is used to provide real-time insights into application behavior, network performance, and user experience, which helps organizations keep their cloud environment resilient and adaptive by enabling:

- **Performance Monitoring:** APM tools monitor performance continuously, and while doing this, they can spot bottlenecks and anomalies prior to affecting end users. This ensures minimal downtime and continuity of service so that quality of service is assured during migration.

- **Resource Allocation and Optimization:** APM correlates application performance metrics with network and infrastructure usage to allocate all necessary resources optimally. This keeps workloads from becoming unruly when demands change.
- **Strategic Decision-Making:** APM data supports long-term planning by linking performance metrics to business outcomes. Organizations can optimize resource allocation, scale intelligently, and identify areas for future investment based on real-time and historical insights.

APM acts as a real-time metaphorical guidepost that keeps applications in line to stay within the confines of zero trust architecture.

ADM and APM: Synergizing Security and Performance

The integration of ADM and APM provides a powerful framework for secure and efficient cloud migration: ADM maps dependencies and designs microsegmentation policies, ensuring secure communication and compliance with zero trust principles, while APM monitors and optimizes performance within those boundaries, validating that segmentation does not impede operational efficiency. This synergy allows organizations to enhance migration precision because ADM ensures that all critical dependencies are identified and migrated together, and APM validates that workloads perform as expected during and after the transition. Additionally, it helps implement adaptive security because ADM facilitates the creation of microsegmented environments, and APM ensures that performance remains optimal within these segments. Finally, it drives continuous improvement through real-time insights from APM that enable ongoing optimization of network traffic, resource utilization, and security policies, informed by the comprehensive dependency maps created by ADM.

Leveraging Hyperscaler Observability Tools for Strategic Cloud Integration

Integrating ADM and APM with cloud-native observability tools from major hyperscalers provides a robust framework for secure, high-performance cloud operations. These tools enhance visibility, performance optimization, and the enforcement of zero trust principles.

Cloud-Native Observability Enhancing ADM and APM

- **AWS**
  - **Amazon CloudWatch:** Offers unified monitoring for AWS resources and hybrid environments, complementing ADM by identifying interdependencies and supporting APM with detailed telemetry to track performance trends.
  - **AWS X-Ray:** Provides end-to-end tracing of application requests, aiding ADM in mapping dependencies and validating microsegmentation policies.
- Azure
  - **Azure Monitor:** Aligns with APM by collecting telemetry from applications and infrastructure, enabling performance optimization and resource allocation.
  - **Azure Network Watcher:** Complements ADM by diagnosing network health and ensuring compliance with segmentation and zero trust principles.
- Google Cloud
  - **Cloud Monitoring and Logging:** Strengthens APM with real-time diagnostics of application uptime and performance while validating the impact of ADM-driven segmentation strategies.
  - **Network Intelligence Center:** Extends ADM capabilities with unified network monitoring, ensuring secure and efficient connectivity in hybrid and multicloud deployments.

Core Benefits of ADM and APM Integration in Dynamic Cloud Environments

Pairing application dependency mapping and application performance management with hyperscaler observability tools adds the perfect holistic solution for secure, well-orchestrated, and responsive cloud operations. Not only do these tools augment visibility and performance, but they also allow for continuous evaluation and dynamic adaptation in changing cloud landscapes.

- **Enhanced Visibility:** ADM quickly maps the complicated interdependencies between applications, networks, and infrastructure in hybrid and multicloud environments. With complementary hyperscaler observability tools, the combination helps to gain real-time visibility into application behavior, resource usage, and network performance, ensuring segmentation approaches are accurate and actionable.
- **Active Optimization of Performance:** APM monitors both app and network metrics in real time; it finds bottlenecks, anomalies, and inefficiencies before they start to cause problems. Using observability tools around such as Amazon CloudWatch or Azure Monitor, businesses can continuously fine-tune application, resource allocation, and workload performance in the context of operational stability.
- **Dynamic Adaptation to Change:** ADM and APM enable organizations to adapt to changing dependencies, workloads and traffic trends in a dynamic manner. Based on continuous updates to dependency maps and real-time performance data, segmentation policies and resource allocations can remain aligned with zero trust principles and operational needs.
- **Confirmed Security and Stability:** Hyperscaler observability tools validate the reliability of microsegmentation, access controls, and general compliance with the principles of zero trust. APM, on the other hand, ensures that applications work within these boundaries, limiting risk, while maximizing user experience and service quality.

Organizations can achieve the following when integrating **Application Dependency Mapping (ADM)** and **Application Performance Management (APM)** with hyperscaler observability tools:

- **Low Risk Migration:** All dependency mapping, along with segmentation design, vastly reduces risk during migration. You can be confident that your most critical workloads can move, secure and sound.
- **Zero Trust:** It is based on strict segmentation and access controls, often based on real-time insights drawn from observability tools.
- **Operational Excellence:** Being constantly monitored and optimized, resources are used properly, ensuring that scaling and resiliency are maintained.
- **Resilient Cloud Architectures**: Integrating ADM and APM data with hyperscaler observability tools supports **adaptive security**, **high performance**, and **actionable intelligence** for long-term architecture planning.

This cohesive approach embeds security and performance at every layer of the cloud journey, providing the resilience that organizations need to succeed in today’s hybrid, multicloud world. With zero trust at the design and implementation processes of cloud architecture, ADM and APM lay the foundations for scalable, secure, and pragmatic cloud architectures, while ensuring operational excellence.

Integrating Observability into Cloud Migration

The primary purpose of observability architecture is to gain an understanding of how applications behave, from the initial development stages all the way to production. It does this by creating ultra-fine-grained behavioral insights of the systems with tons of context from different sources where data is needed. It provides the foundation that is needed in keeping applications functioning at the speed required by new applications architectures (microservices, serverless, Infrastructure as Code) that require new tools to monitor and manage.

With the move to DevOps and the never-ending cycles of rapidly building, testing, and deploying in agile environments, it has become more and more important to be able to monitor applications that have been rolled into production, quickly and easily. These requirements are exactly what observability architecture solves by providing an end-to-end observability framework to monitor modern tech stack.

Key Outcomes

Observability provides several critical outcomes:

- **Comprehensive Data Gathering:** Data pertaining to cloud infrastructure resources, hybrid, and on-premises apps.
- **Unified Platform:** A consolidated location for gathering performance and operational data through logs and metrics, providing an alternative to point system and app monitoring silos.
- **Tech Stack Monitoring:** Coverage of applications, infrastructure, and services.
- **Automated Actions:** Alarms, logs, and event data to improve mean time to resolution (MTTR).
- **Actionable Insights:** Insights to improve the performance of an application and control its resources. You can gain insight into general operational health across the whole system and by individual components.

Metrics and Monitoring

Organizations assess how effective systems and processes are at meeting these service levels by querying monitoring tools for this data. CPU utilization and disk reads of compute instances can provide insight into how these metrics can assist with autoscaling workloads to accommodate higher loads, identifying debugging problems, or simply gaining more insight about system behaviors.

Here are some of the most commonly monitored metrics:

- Availability and latency

- Downtime and uptime of applications
- Completed transactions
- Operations that are successful and failed
- Metrics such as sales numbers and engagement that are the KPIs

The Monitoring service also uses alarms, and a Notifications service defines triggers to notify when metrics cross the defined thresholds. These alarms help organizations keep aware of the problems constantly and revert to the OK status or return the counters to 0 after the situation is restored. Configuring repeat notifications to remind stakeholders of the lack of resolution of ongoing problems can also improve response time and overall service experience.

The Role of OpenTelemetry and CNCF in Cloud Migration

Using the concept of OpenTelemetry (OTel) and Cloud Native Computing Foundation (CNCF) frameworks can improve the strategies of measuring network performance, especially in the context of cloud migration. OpenTelemetry provides a single way to collect traces, metrics, and logs, which makes it an obvious choice for observing and monitoring distributed systems. Pair it with CNCF frameworks that are meant to create agile, scalable, and resilient cloud-native applications, and organizations can establish a holistic approach to monitoring that extends beyond network performance to overall application health.

In addition to traditional observability practices, pairing these with CNCF projects is a good way to achieve both out-of-the-box and cost-effective observability. These CNCF frameworks integrate perfectly with hyperscaler observability tools to make a flexible, scalable approach to monitoring possible.

The architecture shown in Figure 19-3 is an example of how open-source and open APIs can facilitate the observability pipeline by creating a single context pipeline that collects telemetry data. The CNCF hosts many of the projects that drive the cloud-native application development and operations, for monitoring and observability. These are often part of an overall monitoring strategy that is aligned to cloud migration planning and execution. Some common examples include

- **Application for Metric Collection: Prometheus:** A CNCF graduated project, Prometheus can be used to collect and store the metrics for both network and application. Its querying language is robust, and with alerting, it enables you to define conditions that, when the specified network conditions are met, alert operators to potential network issues.

- **Fluentd for Log Aggregation:** A project under the umbrella of CNCF, Fluentd acts as an aggregator to collect logs from network devices and applications and also forwards logs to a common central logging database. It is essential for pattern recognition that may be indicative of a throughput bottleneck or a possible security issue.

- **Tracing with Jaeger:** Jaeger is a CNCF graduated project that provides a distributed tracing platform intended for the monitoring and troubleshooting of transactions in complex, microservice-distributed systems. It allows operations teams to track requests as they move through various services and identify latency or bottlenecks in the application stack at a specific point. Jaeger works well with OpenTelemetry; it is a backend for storing and visualizing tracing data, which works especially well at root cause analysis of microservices architectures.

- **Visualization with Grafana:** Perhaps one of most known and popular CNCF projects for observability, Grafana is a visualization and dashboarding suite capable of visualizing and analyzing telemetry data from a variety of different sources such as Prometheus or OpenTelemetry natively. It provides operators an interactive dashboarding solution for metrics, logs, and traces to easily visualize application and network health over time. With support for advanced visualizations like heatmaps, graphs, and alerts, Grafana is a must-have for aligning application performance with network performance.

- **Open APIs for Integration:** APIs bridge CNCF tools and hyperscaler platforms, ensuring flexible and adaptable monitoring strategies.
- **Cross-Layer Observability:** CNCF frameworks can now be instrumented with OpenTelemetry or digested using the OpenTelemetry framework, allowing operations teams to progress from the disconnected view of components in a distributed application to cross-layer observability. This method gives a high-level view of the relationship between layers of the application stack, such as infrastructure, services, and application layers, and how they affect the overall health and experience of the system. Augmenting this framework with network telemetry enables teams to understand the impact of network performance on application performance, service availability, and user experience. Such cross-layer observability is essential to successful cloud migration as well as cloud operations optimization.

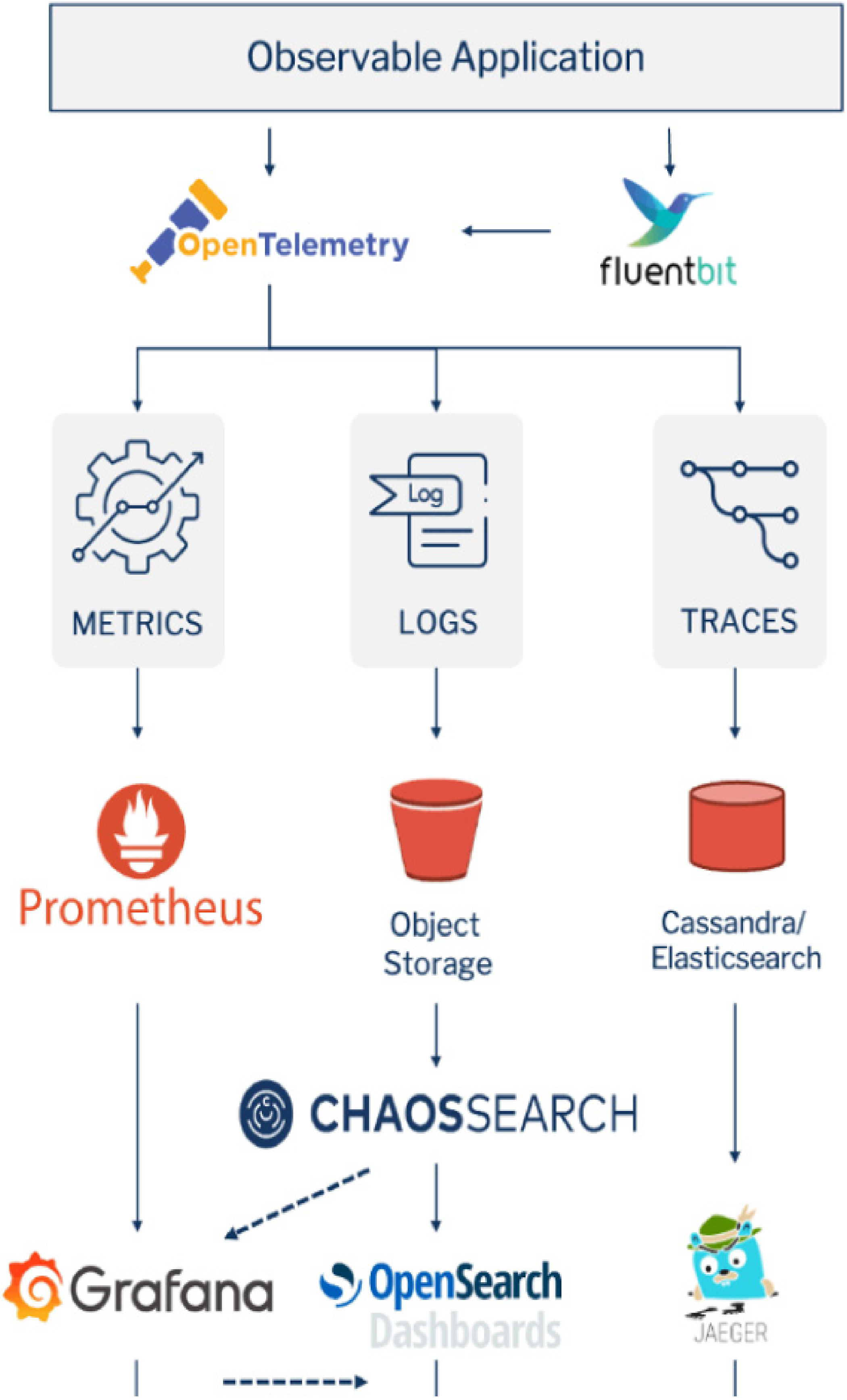

Observable Application
OpenTelemetry
fluentbit
METRICS
LOGS
Log
TRACES
Prometheus
Object Storage
Cassandra/ Elasticsearch
CHAOSSEARCH
Grafana
OpenSearch Dashboards
JAEGER

Figure 19-3 *A Cost-Effective CNCF-Based Observability Architecture, Leveraging Open APIs for Flexibility and Scalability*

It minimizes dependencies on proprietary monitoring tools, providing both vendor and application environment independence and suitability for modern application environments, resulting in a cost-effective approach. By utilizing hyperscaler tools like AWS CloudWatch or Azure Monitor in combination with CNCF frameworks, organizations can maintain the principles of zero trust and achieve a unified observability strategy.

Integrating OpenTelemetry into Network Measurements

Observability is expanded with network and application monitoring supported by OpenTelemetry and CNCF frameworks. They play a role in embedding observability best practices over network performance, which helps in comprehensive pre-and post-cloud migration monitoring.

OpenTelemetry is a suite of APIs, libraries, agents, and instrumentation that you can use to collect and export telemetry data (metrics logs and traces) in a vendor-agnostic manner. That means regardless of cloud provider, regardless of underlying infrastructure, you can follow a consistent methodology for collecting and analyzing data about the performance of your network and the behavior of your applications. This integration can assist in numerous ways, namely:

- **Unified Data Collection:** By integrating OpenTelemetry with your network performance tools, such as TRex or hyperscaler-specific monitoring tools, you can collect comprehensive telemetry data across your entire stack, from network layer metrics to application performance indicators. OpenTelemetry is also capable of integration through traditional networking protocols such as SNMP, Syslog, and Model-Driven Telemetry (Cisco MDT), just to name a few.

- **Enhanced Visibility:** OpenTelemetry's ability to capture traces and metrics from applications can complement traditional network performance metrics, offering deeper insights into how application behavior impacts network usage and performance. This can be an alternative for expensive APM/NPM integrations to achieve similar visibility.
- **Cross-Platform Compatibility:** Since OpenTelemetry works across cloud environments and is supported by major hyperscalers and CNCF projects, it enables consistent monitoring practices whether your workloads are on-premises, in a single cloud, or spread across multiple clouds.

Integration Approaches

OpenTelemetry (OTel) itself is designed primarily for application-level telemetry data—namely, traces, metrics, and logs. Therefore, networking data should be integrated carefully and methodically as well. For network-level data, organizations may rely on support from networking gear or an external software layer that can convert network metrics into OTel-compatible formats. However, some options do exist:

- **On-Premises Networking Equipment:**
  - **Network Device Exporters:** When networking equipment doesn't natively support OTel, you can utilize or implement exporters that scrape metrics from these devices and export them to OTel in a compatible format. They can extract Simple Network Management Protocol (SNMP) data, Syslog messages, or other telemetry data output from the network device and map this telemetry to OTel metrics. An example is referenced in detail in the "Accelerating Your Cloud-Native Observability via OpenTelemetry" workshop described in the following section.
  - **Software Agents:** You an install software agents in network devices (depending on vendor support) or on intermediary systems to gather network performance data and send it to OTel collectors. These agents can collect data on throughput, latency, jitter, error rates, and so on.

  - **Custom Integrations/Proprietary Device Metrics:** Some metrics from proprietary network devices are not readily available in OTel, so you will need to write custom scripts or applications. Such integrations pull data from the APIs or CLI of the network devices and push that data to the OTel collectors.
- **Cloud-Based Measurements:**
  - **Cloud Provider Tools:** Major cloud providers offer various tools and services that monitor network performance within their environments (such as AWS CloudWatch, Azure Monitor, and Google Cloud's operations suite). While these tools don't export data directly to OTel, you can use available APIs or integration services to pull metrics from these tools into OTel.
  - **Cloud Provider Metrics Exporters:** Some open-source projects and third-party solutions have been built to serve as a bridge between cloud provider monitoring tools and OTel. These exporters are able to scrape network metrics (such as VPC flow logs, network latency, and packet loss metrics) and export them in OTel format.
  - **Serverless Functions:** To trigger on specific network events or at a set interval to collect network performance data, you can deploy serverless functions in your cloud environment. These functions can then forward the data to OTel collectors.
- **General Considerations:**
  - **Alignment of Network Metrics:** The network metrics collected from different locations should be aligned in terms of their names, measurements, and granularity so that unified analysis and monitoring happen across your infrastructure.
  - **Security and Compliance:** When collecting and transmitting network performance data, especially in regulated environments, you must ensure that data handling and transmission comply with relevant security standards and regulations.
  - **Scalability:** You must take into account how scalable your solution is, particularly in fluid cloud environments where network topology

and workloads can shift. It should be flexible enough to grow and change without needing a complete overhaul.

Integrating networking data into OpenTelemetry offers a comprehensive view of your infrastructure's performance, combining application and network telemetry for a holistic understanding of system behavior and user experience. While this integration may require additional steps or custom solutions, the benefits of a unified telemetry platform that encompasses both application and network performance metrics are substantial, offering deeper insights and more effective observability across your entire IT ecosystem.

A Real-World Example for Integrating OpenTelemetry into the Networking Domain

Created by the CX-CTO Team of EMEA for Cisco Live 2024, the "Accelerating Your Cloud-Native Observability via OpenTelemetry" workshop is a hands-on educational session designed for network operators to explore integrating OpenTelemetry into network observability strategies. Attendees learned the basics of OTel, its place in improving observability throughout cloud-native environments, and how to fast-track an OTel journey with Cisco's tools. The workshop included hands-on activities, such as streaming telemetry (using Cisco's Model-Driven Telemetry [MDT]), configuring SNMP to Prometheus, and Syslog-events forwarding to OpenSearch.

With these techniques and some simple cloud-native tools, the attendees were able to connect traditional monitoring with modern observability to create vendor-agnostic telemetry pipelines and to perform real-time network performance monitoring. (For more information and access to the workshop's content, see "Cisco GitHub Repository for the Workshop" and "Guide to Converging Infrastructure Monitoring and Observability Using OpenTelemetry" in the "References" section at the end of this chapter.)

Tying It All Together

Let's consider a 360° monitoring strategy: Complementing OpenTelemetry-based monitoring with network performance measurement tools is essential for migrating to the cloud and managing the cloud successfully. In this way, you not only will be able to detect the network performance challenges and resolve them on the go but will also get context as to how these network performance challenges are impacting application performance and the end-user experience. With consistent monitoring capabilities and open-source, cloud-agnostic tooling, organizations will be able to guarantee that their cloud environments have the potential for high performance, resiliency, and security. This provides a comprehensive insight into the infrastructure and application layers, enabling informed decision-making and strategic planning for future growth and optimization in the cloud.

Hyperscaler observability tools are usually very powerful and, as a result, come with a certain cost associated with them. CNCF frameworks focus on cost efficiency, which makes it easy to integrate cost-efficiency with powerful hyperscaler observability tools to find an equilibrium between broad visibility and costs.

The architectural design illustrated in this chapter serves as a guidepost for businesses aiming to build an observability solution that aligns with zero trust principles, ensuring data integrity, resource efficiency, and operational excellence.

End-to-End Visibility on Digital Experience Using Cisco ThousandEyes

Cloud migration is like a grand adventure where the biggest challenge is executing all the moving parts so that end users do not experience any disruption in their digital experience. The conductor of this orchestra is end-to-end visibility, an idea that represents the pillar for IT teams sailing through the turbulent waters of service migration between platforms. Tools like Cisco ThousandEyes can not only light up the path but also ensure that each note is played to perfection.

The Prologue: Setting the Stage for Migration

Picture yourself embarking on a journey across an unknown landscape. Your task: to migrate a wealth of critical services from the comfort of a venerated legacy platform to the great but unknown cloud beyond. The stakes? Ensuring that the end-user experience remains uninterrupted, preserving productivity, and safeguarding revenue. In this odyssey, running blind is not an option; visibility is your most trusted guide.

Figure 19-4 exemplifies the complexity of modern IT environments, highlighting the intricate interconnections between users and services (see “How End-to-End Visibility Can Help with IT Migration” in the “References” section). This underscores the need for end-to-end visibility.

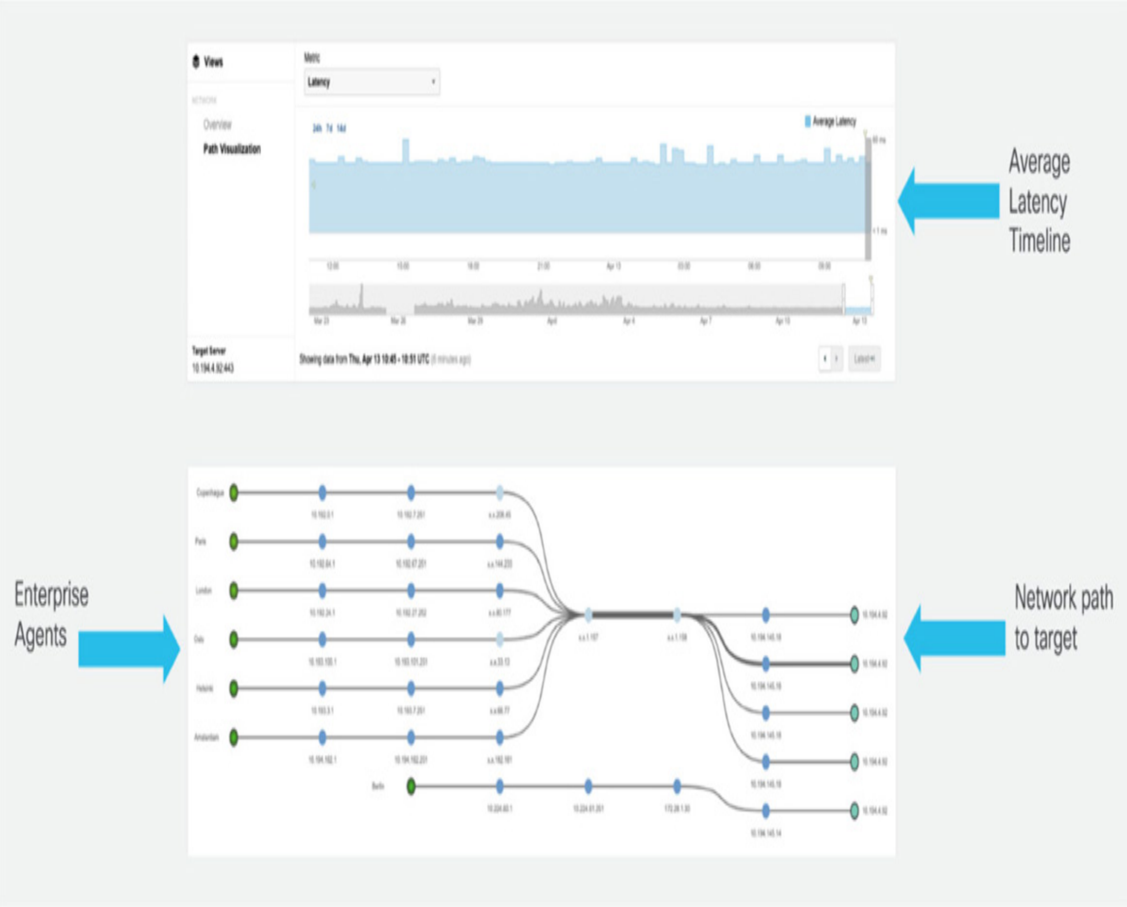


Figure 19-4 *Cisco ThousandEyes Cisco ThousandEyes: End-to-End Visibility on Digital Experience*

Cisco ThousandEyes provides this visibility, offering unparalleled insights into the digital experience and tracking IT service performance before, during, and after migration. With path visualization, IT teams gain a compass for identifying problems as they show up, ensuring everything is calculated and well covered.

The Journey: Navigating the Migration

There are a lot of moving parts to coordinate as the migration takes place; network infrastructure, servers, applications, and data all move to their new abode. The path is fraught with potential disruptions, from increased IT tickets to lost productivity. Here, ThousandEyes plays a critical role, not just as a tool but as a strategic ally, offering insights that transform uncertainties into predictable outcomes. Figure 19-5 demonstrates the layered approach ThousandEyes uses to monitor and optimize performance.

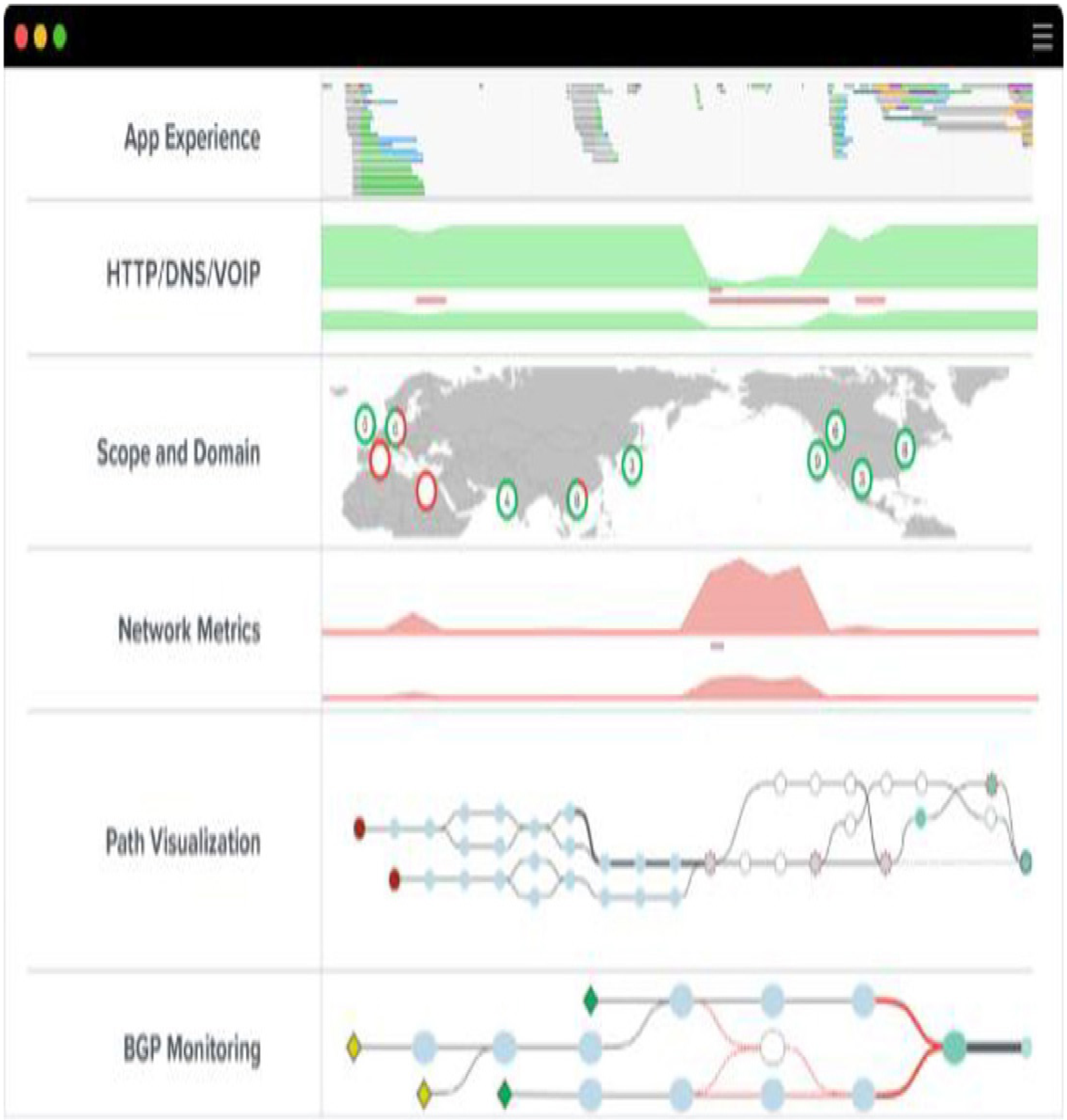


Figure 19-5 *Visualize Data Across Layers in One View (Source: https://www.thousandeyes.com/product/platform)*

By visualizing the flow of data across networks, ThousandEyes empowers IT teams to see through the fog of migration, quickly identifying and resolving issues. This visibility extends across all layers and domains between users and applications or services, ensuring the migration process is precise and well planned.

The Finale: Validating Success

With the migration complete, the curtain rises on a new era. Yet, the performance is not over until success is validated. How do you ensure that the services have not just migrated but thrived in their new environment? How do you demonstrate the migration's success to key stakeholders, backed by incontrovertible evidence?

ThousandEyes provides detailed pre- and post-migration reports, benchmarking the digital experience before migration and measuring it against the post-migration reality, as seen in Figure 19-6 (again, see "How End-to-End Visibility Can Help with IT Migration" in the "References" section).

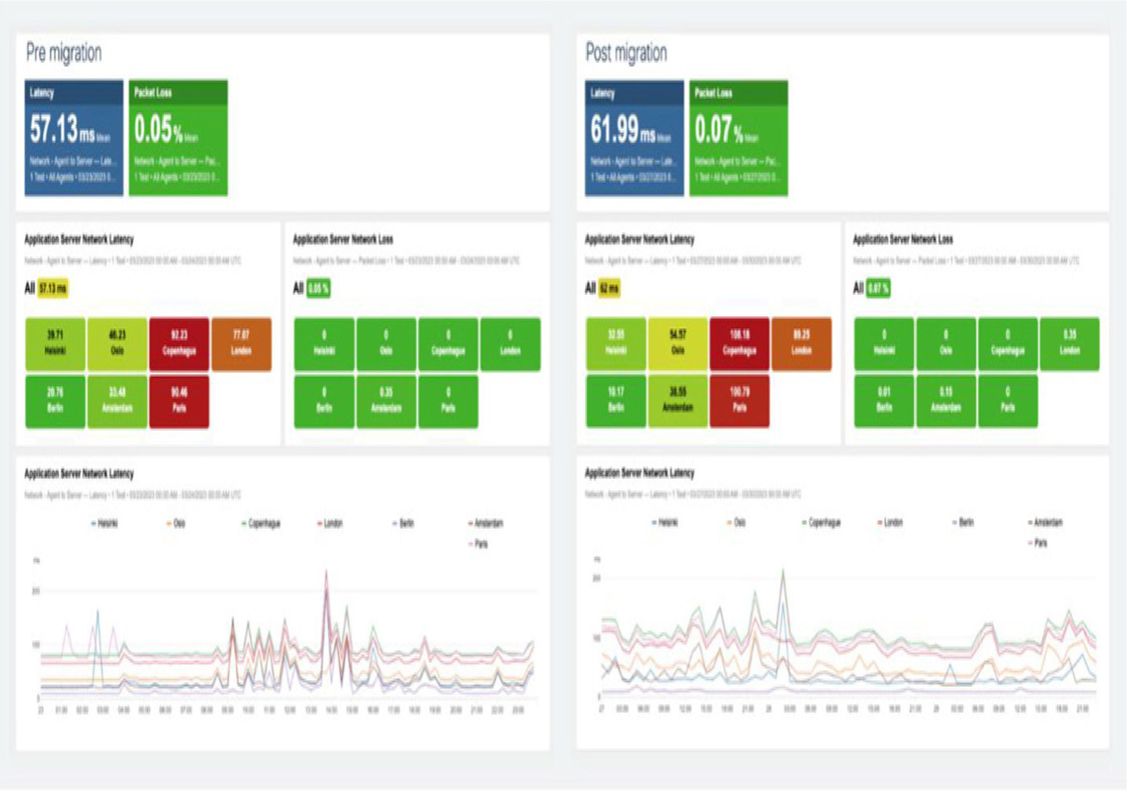


Figure 19-6 *Cisco ThousandEyes: Pre- and Post-Migration Report Example*

These reports highlight areas of triumph and opportunities for improvement. This holistic visibility extends beyond the immediate environment, enabling IT teams to understand the migration's impact across all domains and layers.

ThousandEyes' repeatable monitoring templates not only validate the success of current migrations but also equip IT teams with the insights and tools necessary for future migrations, creating a legacy of visibility that enhances digital transformation efforts.

Managing IP Addressing and DNS Changes

A cloud migration strategy requires careful management of **IP addressing** and **DNS configurations** because maintaining accessibility to applications and services is critical during the migration process. This strategy includes pre-migration assessments, planning, and post-migration configurations to ensure the cloud environment operates smoothly and that applications and services remain accessible once deployed in the cloud.

The challenges and best practices associated with IP planning are pivotal in avoiding conflicts, ensuring scalability, and maintaining resiliency and availability in a cloud or multi-cloud adoption strategy.

- **IP Address Management (IPAM)**: A well-defined IP addressing scheme is essential for your cloud network to avoid conflicts with on-premises IP ranges while providing scalability for cloud resources. If needed, private IP spaces in the cloud can be mapped to public IPs using **Network Address Translation (NAT)**.
- **DNS Updates**: After a successful migration, update DNS records to reflect the new service locations. This may involve updating **A records** to point to new IP addresses or using **CNAME records** for abstraction, simplifying DNS management.
- **DNS Services**: Utilizing **cloud provider DNS services** enhances integration and orchestration in the cloud environment. These

services often provide traffic routing, load balancing, and automatic failover for improved availability and performance.

- **Phased Migration:** For phased migrations, consider implementing **split-horizon DNS** or other techniques to serve different DNS responses based on the origin of the DNS query. This ensures a seamless transition between the old and new environments.

Challenges in IP Planning

Effective IP planning can help companies avoid challenges that may impede operations as they transition their networks to cloud environments:

- **Overlapping IP Addresses:** Such addresses, between cloud and on-premises resources, may lead to conflicts quickly, impacting connectivity.
- **IP Address Management:** Effective management is critical, so you need to assign unique IP addresses to cloud resources, avoiding duplication and routing issues.
- **IP Address Range Allocation:** If certain departments or teams have been assigned pre-existing IP ranges, and those ranges overlap with the IP ranges that will be used when migrating services to the cloud, then they may conflict.
- **Route Aggregation Issues:** Departments may expand and deplete their allocated IP ranges, making route aggregation hard, and sometimes lead to unpredictable assignments of the segments.
- **Scaling and Elasticity:** Due to auto-scaling in a cloud environment, IP allocation needs to be dynamic and in segments, which, in turn, will need prior whitelisting on a firewall.
- **Human Readability:** Good IP segmentation helps to make things more readable for your network admin and makes it easier to troubleshoot faster.

Best Practices for IP Planning

Bringing IP planning into the overall cloud migration plan guarantees that networking functions are transitioned to the cloud without sacrificing communication transparency or system security.

1. **Pre-Migration Assessment:** A comprehensive assessment of existing IP allocations will identify potential conflicts as well as the required range of IP segments that need to be made available in the cloud.
2. **Dedicated IP Address Range:** The creation of a new IP range for cloud workloads reduces the likelihood of a clash between cloud and on-premises resources.
3. **IP Address Management Tool:** You should use a single tool across the organization to manage and track an IP address, including those allocated for the cloud resources.
4. **Flexible IP Address Ranges**: Using private IP ranges along with **Network Address Translation (NAT)** allows for more flexible and efficient IP utilization. This approach simplifies IP planning and provides scalability for future maintenance and expansion needs.
5. **Automatic IP Address Assignment:** Utilizing cloud capabilities for automatic IP allocation enables infrastructure scaling and minimizes the need for manual configuration.
6. **IP Address Audits:** Network administrators periodically audit IP addresses to detect overlap or collision, thus maintaining the integrity of the IP addressing scheme.

Designing Virtual Networks

Building virtual networks not only will help in securing the workloads over the cloud but also ensure the efficiency and scalability of the resources. These networks should be logically isolated to allow direct, private communication between resources, yet be capable of supporting both private and public IP addresses. The goal is to strike a balance that leaves an environment that is secure but that can adapt to the ever-changing nature of cloud services. Virtual networks are not just about the tech but about the high-level goals of cloud migration, ensuring security, scalability, and resilience in the real world. Here are some other design points to consider:

- **Logical Isolation for Security and Performance:** The principle of logical isolation is the key to virtual network design for security and performance. This strategy allows each network segment or resource group to be isolated on the cloud, decreasing points of unauthorized access and lessening points of failure. Logical isolation is part of a defense-in-depth strategy that overlays numerous controls between resources; the purpose is to delay lateral movement by malicious actors as they attempt to move through the infrastructure.

- **Direct and Private Communication:** Resources must be able to reach out to each other by direct and private messaging to ensure the efficient performance of cloud applications. These resources allow for the use of virtual private networks and other networking technologies to establish a secure connection without the need for data to go across the public Internet. These are important to ensure that only identifiable and authenticated users are allowed to access sensitive applications and data.

- **Integration of Private and Public IP Addresses:** The right mix of private and public IPs in a virtual network creates a secure and accessible virtual space. Cloud resources all communicate internally with private IP addresses, meaning this traffic is hidden from the public eye, which protects it from external threats. Public IP addresses, on the other hand, are dedicated to resources that need to be accessible from the Internet, such as web servers. These addresses must be managed carefully, such as with NAT services, to keep them in a secure perimeter and still be accessible where necessary.

Universal Best Practices Across Cloud Platforms

Although these principles are framed with Azure in mind, they are platform-agnostic best practices that can be applied to any cloud (AWS, Google Cloud Platform, etc.). The fundamental notions of isolation, secure communication, and IP address management are essential in constructing virtual networks that facilitate the migration of cloud workloads across clouds.

Practically speaking, implementing these universal best practices involves:

1. **Creating Segmented Network Zones**: Divide virtual networks into zones based on function, sensitivity, and exposure level. This segmentation allows appropriate security policies and controls to be applied according to the requirements of each zone.
2. **Leveraging Subnets for Detailed Control**: Virtual networks can be further segmented into subnets for granular control of traffic flow and access rules. Applying security policies at the subnet level enhances security posture by creating layered defenses.
3. **Designing for Scalability and Flexibility**: Plan for future growth and changes in the network design to ensure that the virtual network can scale seamlessly with the organization's needs. This involves planning IP address space allocation, choosing appropriate network topologies, and considering integration with other cloud services.
4. **Utilizing Security Groups for Access Control**: Apply **Network Security Groups (NSGs)** and **Application Security Groups (ASGs)** to define fine-grained access control policies. NSGs allow inbound and outbound traffic rules at the resource level, while ASGs group resources with similar security needs, enabling dynamic policy management. (*Detailed explanations are provided in the Security Considerations section*.)

These best practices enable you to build virtual networks that not only meet the technical requirements of cloud applications but also provide a high level of security and performance. By applying these principles, organizations can construct zero-trust architecture-based networks that verify each request, provide least-privilege access, and limit communication at every level. Ensuring that nothing is trusted and everything is verified constantly transforms the network into a secure and resilient defense against evolving cyber threats.

Best Practices for Virtual Networks Designs and Considerations During Cloud Migration

Incorporating the following best practices into your cloud migration strategy ensures that your virtual network infrastructure will be robust, secure, and capable of supporting your organization's dynamic needs. Applying these best practices as you build out your cloud migration plan guarantees your virtual network architecture will be strong, secure, and can handle the ever-changing demands of your business.

- **Network Design Fundamentals**

  Virtual Network Design**:** You can create private, isolated networks specific to your subscription to allow the various Azure resources to interact privately and securely. This practice provides isolation between the virtual networks, so any virtual machine host vulnerabilities cannot cross networks.

  IP Address Management (IPAM)**:** Planning the IP space for your network carefully is important so that there are no conflicts with on-premises networks and you can scale seamlessly in the future. Each virtual network should have a CIDR range of no more than /16 to save IP addresses for efficient utilization and divide and conquer the subnets logically.

  Subnet Design**:** Subnetting virtual networks based on technical or organizational requirements provides a more granular level of control over network traffic flow while reducing the broadcast domain, thus improving network security.

- **Connectivity and Scalability**

  Hub-and-Spoke Network Topology: A hub-and-spoke topology isolates workloads while centralizing shared services such as security and connectivity. This design enhances security and manageability by focusing services in the hub while keeping workloads isolated in spokes.

  DNS Configuration: Proper DNS configuration ensures seamless name resolution within and between virtual networks, as well as with on-premises networks. This can be achieved using custom DNS servers or cloud provider DNS services.

Hybrid Cloud Networking: For organizations adopting a hybrid cloud strategy, establishing secure and reliable connectivity between on-premises and cloud environments is essential. Solutions such as Site-to-Site VPNs or Azure ExpressRoute (and equivalents on other clouds) are effective for creating dependable hybrid connections.

- **Security Considerations**

Security Measures: Implement perimeter networks, Network Security Groups (NSGs) for traffic filtering, and services like Azure Firewall to protect cloud resources. These measures form the foundational security layer for cloud-native applications.

Application Security Groups (ASGs): ASGs offer dynamic and scalable options for grouping resources with similar security needs. They simplify security management by allowing policies to be defined at the group level while maintaining fine-grained access control.

- **Reliability and Monitoring**

High Availability and Resiliency: Availability zones protect applications from the loss of an entire data center, while high availability designs safeguard individual applications. Without this approach, continuity of service and data integrity cannot be guaranteed. Therefore, implementing high availability and resiliency measures is fundamental to ensuring reliable and secure cloud operations.

Monitoring Resources and Flow Communications in the Cloud: Monitor communications between virtual machines and endpoints, whether they are other VMs or fully qualified domain names (FQDNs). This practice provides visibility into resource relationships within virtual networks and facilitates troubleshooting of network traffic issues.

DNS

DNS plays a pivotal role in ensuring that applications and services are easily accessible and operate efficiently, especially during and after cloud migrations. At its core, the Domain Name System (DNS) takes human recognizable domain names and converts them into IP addresses that are used for communication between computers. For smooth operation, you need to adapt your DNS strategy to support hybrid areas (on-premises and cloud) when migrating to the cloud. When it comes to cloud migrations, a strategic approach to DNS management is critical for ensuring that cloud-based applications and services can be accessed and connected to without interruption.

Best Practices for DNS Management in Cloud Migrations:

1. **Understand Cloud DNS Capabilities:** Cloud solutions provide a DNS service in which you do not have to maintain your own DNS servers. These services enable both public and private zones for secure, internal communication behind virtual private clouds and expose only specific resources to the public Internet.

2. **Create Consistent Naming Standards:** You should establish a simple, yet consistent, naming convention across your organization. This could involve using separate subdomains for different environments (for example, corp.example.com for on-premises and cloud.example.com for cloud resources) to ease management and enhance security through clear segmentation.

3. **Plan for Hybrid Environments:** In hybrid environments (if you still have some resources running in the data center), DNS resolution has to be reachable across environments. Best practices usually suggest a hybrid model with both cloud-based and on-premises DNS systems. This method enables you to have authoritative DNS resolution within respective environments for efficient and secure name resolution.

4. **Leverage DNS Forwarding:** You should implement DNS forwarding to enable seamless queries between cloud and on-premises environments. This allows you to keep your internal DNS

names resolvable throughout your infrastructure, supporting compatibility and connectivity without hindering security or performance.

5. **Adopt Private Zones for Internal Resources:** Use private DNS zones for resources that are not intended to be public and should be visible only to certain VPC networks. This practice provides internal security, which lessens exposure to external threats.
6. **Configure DNS Resolution Locations:** You must decide where DNS resolution will occur—whether keeping it on-premises, moving entirely to the cloud, or adopting a hybrid model. Each option has its own trade-offs when it comes to latency, reliability, and how the DNS function will be controlled. For most organizations, the balanced approach that the hybrid model provides is often the best option.
7. **Automate DNS Management:** You can use automation tools to manage DNS configurations, especially when dealing with dynamic cloud environments. Automating adjustments makes it easy to update DNS records as necessary while minimizing the chances of human error that can lead to downtimes, project delays, and other negative business outcomes, all while also supporting scalable operations.
8. **Implement High Availability Strategies:** For critical connections, you should consider using services like site-to-site VPNs or dedicated connectivity options (equivalent to Azure ExpressRoute in other clouds) to ensure DNS is highly available and serves its hybrid network in a reliable manner.
9. **Secure Your DNS Architecture:** You can configure security measures like firewall rules and access controls to protect your DNS architecture. To do so, open the firewall for DNS traffic and ensure on-premises DNS and cloud DNS are in sync to avoid unwanted access.
10. **Consider Future Scalability and Interoperability:** You should design your DNS architecture with future growth in mind, ensuring it can accommodate additional resources, cloud services, and

potential multicloud strategies without requiring significant reconfiguration.

Ensuring High Availability and Disaster Recovery Readiness

High availability and disaster recovery readiness are fundamental components of a successful cloud migration strategy. They complement each other to help applications and services be resilient and reliable in the face of potential downtime and data loss. These principles are in line with zero trust architecture because they embed continuous verification, segmentation, and least privilege in each step of the process. The following sections explain how you can create a solid HA and DR posture for your workloads in the cloud—starting from pre-migration assessments, selecting the right cloud provider and service model, designing the architecture, and executing the migration while ensuring continuous monitoring and optimization.

Before your organization migrates to the cloud, it is essential to assess your current disaster recovery capabilities. This process begins with identifying critical business processes, applications, and data while defining recovery objectives. Recovery point objective (RPO) measures the maximum acceptable amount of data loss in the event of a disruption, while recovery time objective (RTO) defines the maximum downtime your business can tolerate. These metrics provide the foundation for designing an effective DR strategy.

Continuous assessment of risk and vulnerability is a key tenet of zero trust, and this perfectly correlates to evaluating your current backup and recovery solutions. New systems must be assessed for their reliability, security, cost-effectiveness, and frequency of testing. You should also consider if the solutions in place strictly follow access control policy along with encryption standards as they must in order to protect the integrity of the data from any disruption that it may face. Deciding on how these solutions line up with your organizational business objectives and your agenda for cloud functionality will ensure they hold up to the rigorous verification within zero trust.

Selecting a Cloud Provider and Service Model

Selecting a cloud provider and a service model for disaster recovery planning is a critical task. Features, pricing model, and service-level agreements for disaster recovery vary from provider to provider. For example, some providers include built-in backup tools and geographic redundancy, whereas others require third-party solutions for these capabilities. Popular considerations include

Popular considerations include:

- **Failure Tolerance**: The ability of a system to continue operating even when one or more of its components fail, ensuring overall system resilience. This is different from **fault tolerance**, which focuses on handling internal errors without impacting functionality.
- **Availability Zone/Region Redundancy**: Ensures that data and applications remain accessible even if an entire data center or region goes down.
- **RPO (Recovery Point Objective) and RTO (Recovery Time Objective)**: Evaluates whether the provider can meet the required recovery metrics at a cost that aligns with budget constraints.

Your disaster recovery model is also influenced by the choice of service model such as Infrastructure as a Service, Platform as a Service, or Software as a Service. The application of zero trust principles, though, deepens the emphasis on evaluating each model type in terms of security posture. For instance, IaaS gives the maximum level of control over disaster recovery infrastructure but brings the need for strong adherence to access control policies and routine verification of data protection mechanisms. Organizations need to keep their eyes firmly focused on the provider's behavior with regard to zero trust (for example, encryption, access segmentation, DR process workflows visibility) with any increase in responsibility taken on by the provider for backup and recovery in PaaS and SaaS models.

Organizations can choose a cloud provider and service model that is reflective of zero trust principles, where secure access, limited exposure, and continuous validation of the cloud infrastructure are bedrocks of disaster recovery.

Designing Your Cloud Architecture and Migration Strategy

After you select the provider and service model, the next step is to design the cloud architecture and migration strategy as per disaster recovery preparedness. Start with selecting the right migration approach (such as lift-and-shift, replatforming, or refactoring) based on complexity, compatibility, and performance. All these methods have their own advantages and drawbacks (as covered in the previous sections of the chapter); thus, the decision to use any specific method should reduce the risk and disruption, while maximizing the performance of workloads in cloud.

Redundancy and resilience are both foundational aspects of zero trust; therefore, a robust cloud architecture must also prioritize redundancy and resilience. Critical components should be spread across availability zones or regions for high availability while resource segmentation prevents lateral movement when a compromise occurs. You should treat traffic borders between zones as strong barriers and use load balancers to balance traffic on multiple instances, allowing for uptime and performance, while maintaining strong boundaries between zones. Coupled with zero trust automation, automated failover makes rapid recovery possible with minimum human intervention, which strengthens the feature of resilience against outages.

Planning for cloud-based backup and recovery solutions is equally important. These will need to be compliant and secure, but they should also fit your RPO and RTO needs, which means that they will require encryption, least-privilege access, and monitoring at all times. The zero trust-based disaster recovery solution would enforce data and other segmentation for replication, ensure secure replication across regions, and verify access for replication for recovery processes. Moreover, automation is also encouraged with zero trust to constantly verify backup integrity and allow only authenticated entities to restore mission-critical data during recovery and without affecting business continuity.

Executing and Monitoring Migration and Disaster Recovery

The final stage of the process is the execution and ongoing monitoring of your migration and disaster recovery strategies. During migration, it is critical to follow the plan closely, ensuring that data, applications, and infrastructure are securely transferred to the cloud without loss or corruption. Post-migration, workloads must be tested to verify that they meet functionality, performance, and operational requirements.

The core aspect of ensuring that disaster recovery processes align with the spirit of zero trust is monitoring and testing. Disaster recovery solutions must be thoroughly tested to ensure RPO and RTO objectives are being met, while monitoring systems need to constantly monitor backups, recovery points, and recovery times. Under a zero trust approach, constant drills and audits are carried out to identify any loopholes and ascertain preparedness for real-world disruptions. DR strategies should evolve over time to meet new threats, new technologies, and new business needs.

In addition, continuous DR process optimization enforces least-privilege access to DR systems to ensure that only authenticated and authorized personnel can execute recovery actions. By embedding these zero trust principles into execution and monitoring, organizations can maintain a resilient, secure, and adaptable cloud environment.

Security Posture Adjustment Post-Migration

When organizations move substantial IT resources to the cloud, they must take strategic steps to adjust their security posture as well. Shifting security control from on-premises to the cloud has much more to do with work culture rather than simply lifting and shifting security control from one to another. The following sections explore the crucial steps and considerations for adjusting security controls, monitoring and responding to security events during the stabilization period, and updating incident response and forensic capabilities post-migration.

Adjusting Security Controls to the Cloud Environment

Such shift to a cloud environment requires a full hard look at the current security controls and tweaking them. Cloud environments, however, are far from traditional on-premises settings and provide dynamic scalability and flexibility—both of which can be advantageous but can also bring a new set of security challenges and new attack surface.

- **Re-evaluate Security Controls:** This step starts by performing a full audit on your current security controls and figuring out their relevance in a cloud mindset. It also may lead to the decline of some on-premises security measures because they no longer fit neatly in the cloud mold and will either have to change drastically or become irrelevant.
- **Adopt Cloud-Native Security Features:** Most cloud providers come with in-depth security tools and services built into their cloud resource protection. Using these cloud-native controls, however, offers security capabilities that are more tightly coupled with the cloud infrastructure and make for more efficient protection rather than trying to duplicate an on-premises security model.
- **Implement a Least-Privilege Model:** The ephemeral nature of cloud means that strict access controls and least-privilege access is more important than ever. You should modify your IAM policy to make sure that humans and systems have the smallest required permission necessary to execute the actions.
- **Encrypt Data at Rest and in Transit:** All data stored in the cloud should be encrypted (at rest) by default, and data exchanged between cloud services should use a secure connection (in transit) to prevent sensitive information from being accessed by someone who is unauthorized to access it.

Monitoring and Responding to Security Events During the Stabilization Period

Typically, the time directly after migrating to the cloud is when this monitoring and response is going to be most needed. During this stabilization phase, the security configuration is optimized, so you need to make sure everything working as expected in the new environment.

- **Enhanced Monitoring:** You can use cloud-native and third-party tools such AWS GuardDuty or Azure Sentinel to detect threats using real-time analysis. These are examples of tools that can detect unauthorized access attempts, nontypical network traffic, or unexpected resource changes in the cloud.
- **Incident Detection and Response Plan Update:** You can adapt incident response plans to take into consideration the few unique characteristics that a cloud brings. This step consists of consuming information from cloud-native logging and monitoring equipment to allow the immediate discovery and cleanup of threats.
- **Continuous Monitoring:** You can use real-time monitoring tools to monitor your cloud environment to ensure that everything is up-to-date with the security of your cloud environment. This step involves the detection of unauthorized access attempts, network traffic patterns, or changes in cloud resources.
- **Feedback Loop for Security Adjustments:** With a feedback loop from monitoring and incident response activities, cloud environments are more agile and have quicker deployment cycles for security updates and patches.

Updating Incident Response and Forensic Capabilities

Migrating to the cloud also requires updating your organization’s incident response (IR) and forensic capabilities to address the nuances of cloud computing. The cloud's dynamic nature requires a shift in incident response and forensic approaches. This adjustment encompasses the following:

- **Collaboration with Cloud Providers:** You need to know your cloud provider's role in incident response and forensics. Define escalation paths for contacting the security team at the provider during an incident. Take advantage of the incident response and forensic resources offered by cloud vendors. That includes what can be said about the role of a provider during an incident, as well as the resources that can be used to investigate.
- **Revised Incident Response Plans:** You can update your IR plans to include cloud-specific scenarios and procedures. Ensure that your response team is trained on cloud architectures and understands how to access and analyze cloud logs and data for incident investigation.
- **Cloud-Specific Forensic Tools:** Traditional forensic tools may not be effective in cloud environments due to the abstracted nature of cloud infrastructure. Invest in cloud-specific forensic tools that can capture and analyze cloud-based evidence without disrupting operations.

Using Hyperscalers and Cisco-Provided Security Tools

As organizations move to the cloud, it is critical to understand the security tools offered by hyperscalers such as AWS, Azure, and GCP, alongside complementary security-focused tools from third parties such as Cisco. These tools are security enhancing, compliance enabling, and incident response and container-based deployment facilitators for cloud infrastructure. Using these tools, companies can protect their assets from all types of cyber threats, adhere to regulatory compliance requirements, and perform fast incident response and forensic analyses.

- **AWS Security Tools**
  - **AWS Shield:** This managed distributed denial-of-service (DDoS) protection service helps secure web applications that are running on AWS. AWS Shield delivers scalable, automatic inline mitigations

that reduce application downtime and latency. For example, it is often used to protect Amazon EC2 instances, Amazon CloudFront distributions, and elastic load balancing (ELB) services.

  - **Amazon GuardDuty:** This threat detection service continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. It makes use of machine learning, anomaly detection, and always-on threat intelligence to detect commonly signed and continued threats. For example, GuardDuty can identify compromised instances or instances performing reconnaissance.

  - **AWS WAF (Web Application Firewall):** This tool provides a flexible rules engine to filter web traffic based on conditions that you define, supports control of access to your content, and looks for patterns that match SQL injection or cross-site scripting. AWS WAF provides users with the flexibility to define rule sets that work on both blacklist rules, for guarding against common attack patterns such as SQL injection or cross-site scripting, and whitelist rules that safeguard access to their APIs by setting up a rate rule.

- **Azure Security Tools**

  - **Azure Sentinel:** This tool is a scalable, cloud-native, SIEM and SOAR solution. Azure Sentinel delivers intelligent security analytics and threat intelligence across the enterprise, providing a single solution for alert detection, threat visibility, proactive hunting, and threat response. It is capable of fetching data that is distributed across all users, devices, applications, and infrastructure, on-premises or in various clouds.

  - **Azure Security Center (Microsoft Defender for Cloud):** This tool provides unified security management and advanced threat protection for the hybrid cloud workloads in Azure, on-premises, and in other clouds. It enhances the security posture of data centers and brings leading threat protection to SaaS applications, whether they’re inside or outside Azure, and to on-premises workloads.

- **Google Cloud Platform (GCP) Security Tools**

  - **Google Cloud Armor:** This web application firewall provides threat and DDoS protection for applications and websites. It is designed to protect Google Cloud services like Google Cloud Load Balancing and can be used to create custom rules for traffic filtering, rate limiting, and IP blacklisting or whitelisting.

  - **Google Cloud Security Command Center:** This tool offers comprehensive visibility into and control over cloud resources. It helps identify vulnerabilities, detects threats, performs security assessments, and ensures compliance across Google Cloud services. The tool can, for example, find storage buckets that are publicly accessible, detect instances of cryptocurrency mining, and identify misconfigurations.

- **Cisco Security Tools**

  - **Cisco Secure Cloud Analytics (formerly Stealthwatch Cloud):** This tool offers unparalleled visibility and security analytics across your cloud and on-premises environments. It uses behavioral modeling, machine learning, and global threat intelligence to detect advanced threats. Secure Cloud Analytics can quickly identify malicious activity, such as unusual data exfiltration attempts or communication with known bad domains.

  - **Cisco Duo:** This user-friendly access security tool provides two-factor authentication, endpoint security, and secure single sign-on capabilities for both cloud and on-premises applications. It ensures that only the right users and secure devices can access applications. For example, Duo can enforce policies that check the security hygiene of devices before granting them access to applications, ensuring that outdated devices don’t become a security liability.

  - **Cisco Umbrella:** This cloud-delivered security service provides safe access to the Internet and use of cloud apps everywhere. It blocks malicious destinations before a connection is ever established, and it offers DNS layer security, secure web gateway, firewall, and Cloud Access Security Broker (CASB) functionality. Cisco Umbrella can protect users from phishing, malware, and ransomware attacks, regardless of their location.

Reconfiguring the security posture after migration is an evergreen process of adaptation and optimization and not a one-time effort. By using cloud-native security solutions, revisiting security controls, performing active monitoring and response during the stabilization phase, and enhancing incident response and forensic capabilities, organizations can secure cloud environments from new threats and adapt to the elastic nature of the cloud. Each one of these is important to the new digital normal and to keeping a strong cyber-postured landscape.

Identity and Access Management in Hybrid Environments

The blend of cloud and on-premises resources has given birth to hybrid environments. These ecosystems, while offering the flexibility and scalability essential for contemporary business needs, also introduce complex challenges in managing access and securing sensitive resources. Among the pillars supporting the security framework in such environments, identity and access management (IAM) stands out as a cornerstone. In a zero trust framework, IAM is not just a tool but a foundation for verifying and enforcing access policies across both cloud and on-premises resources. Hybrid environments benefit from zero trust's focus on granular access controls, ensuring that no implicit trust is granted, regardless of location.

IAM Basics

Identity and access management serves as the cornerstone of any organization's security framework, especially as businesses increasingly adopt hybrid environments that blend cloud services with traditional on-premises infrastructure. At its core, IAM ensures that the right individuals (or entities) have access to the technological resources they need, at the right time and for the right reasons. It encompasses a range of practices and technologies designed to manage digital identities, authenticate users, authorize access to resources, and enforce security policies across an organization's systems and applications.

Significance of IAM in Hybrid Environments

Hybrid environments introduce significant complexity in managing access controls due to the amalgamation of cloud-based resources and on-premises systems. Maintaining a consistent security posture is crucial. Disparate systems often have different security models and IAM capabilities, raising the challenge of enforcing uniform access policies and preventing security loopholes. Adopting a zero trust approach helps organizations to assure that access decisions are made dynamically based on identity, device posture, and contextual risk, which mitigates the potential vulnerabilities introduced by the amalgamation of disparate systems in hybrid environments.

Strategic Considerations for IAM in Hybrid Environments

As organizations migrate to the cloud, rethinking IAM strategies is a must. The process begins with a thorough assessment of existing IAM roles and policies in the on-premises infrastructure to determine which can be directly transferred to the cloud, which need adaptation, and which are no longer necessary. Understanding the IAM models offered by chosen cloud providers is crucial. Cloud services such as AWS, Azure, and GCP each have their unique IAM tools and best practices. Adapting to these models ensures that migrated IAM policies are optimized for cloud environments, leveraging native tools for enhanced security and management.

Hybrid IAM solutions that offer seamless management of identities and access across both cloud and on-premises environments are worth considering. Such solutions support federated access, allowing users to log in once and access resources across multiple environments securely. They also enable the synchronization of identities and roles and unified policy enforcement across the IT landscape.

Hybrid IAM solutions offer a unified view and control mechanism over users’ access rights across all environments, enabling organizations to

- **Simplify User Access:** By providing a single identity for each user who works across both cloud and on-premises environments, reducing the complexity of managing multiple identities and improving user experience.

- **Enhance Security:** Through consistent application of security policies and access controls, regardless of where resources are located, minimizing the risk of unauthorized access and potential security vulnerabilities.
- **Improve Efficiency:** By automating the provisioning, management, and deprovisioning of user access to resources across the hybrid landscape, thereby reducing administrative overhead and improving operational efficiency.
- **Support Compliance Efforts:** With centralized reporting and monitoring capabilities that facilitate compliance with regulatory requirements by providing an audit trail of access and activities across both cloud and on-premises environments.

IAM Role Mapping and Policy Enforcement

At the heart of IAM in hybrid environments lies the critical task of role mapping and policy enforcement. ABC Corp, with its diverse array of cloud and on-premises resources, faced the daunting challenge of ensuring that the right people had the right access to the right resources at the right times and for the right reasons. The solution? A comprehensive role-mapping strategy that aligns with the principle of least privilege.

Role mapping in hybrid environments involves the intricate alignment of user roles across different systems and platforms. For ABC Corp, this meant creating a unified role framework that could be easily translated between its on-premises Active Directory and its cloud-based IAM services. By implementing a central IAM solution that supports role synchronization, ABC Corp was able to enforce consistent access policies across its entire IT ecosystem. This approach not only streamlined access management but also significantly reduced the risk of unauthorized access.

Federated Identity for Seamless Access Control

Federated identity management (FIM) emerged as a game-changer for ABC Corp, offering a seamless access control mechanism across its hybrid landscape. Federated identity leverages shared authentication protocols and standards, such as Security Assertion Markup Language (SAML) and OAuth, to allow users to access multiple applications, both in the cloud and on-premises, using a single set of credentials.

This model of trust between disparate systems enabled ABC Corp to provide employees with a frictionless access experience while maintaining stringent security controls. Employees could now access resources located anywhere in the hybrid environment without the need for multiple logins, significantly enhancing productivity without compromising on security. Moreover, federated identity also facilitated secure collaboration with partners and vendors, extending trust in a controlled manner beyond the organizational boundaries.

Federated identity supports zero trust by enabling seamless access without compromising security, ensuring that every authentication request is validated regardless of location or resource. This strengthens the "verify every request" principle, even when accessing third-party systems or collaborating externally.

Privileged Access Management in a Hybrid Setting

Privileged access management (PAM) is the specialized segment of IAM that focuses on the oversight and control of privileged accounts, which are often targeted by attackers due to their high-level access rights. In a hybrid environment, managing these accounts becomes exponentially more complex. ABC Corp's approach to tackling this challenge was twofold: implementing robust PAM solutions and adopting a zero trust security model.

With PAM solutions that integrate seamlessly with both cloud and on-premises systems in place, ABC Corp was able to enforce strict access controls and monitoring for privileged accounts, regardless of their location. Features such as session recording and just-in-time access provided a detailed audit trail and ensured that privileged access was granted only when necessary and revoked immediately after use.

PAM's integration with zero trust principles ensures that no access, even from privileged accounts, is granted without rigorous verification. By enforcing just-in-time access and continuous monitoring, zero trust principles prevent privilege abuse and reduce the risk of lateral movement by attackers. Moreover, by embracing a zero trust philosophy, ABC Corp ensured that trust was never assumed based on network location. Every access request, whether from a privileged user or not, was subjected to rigorous authentication and authorization, ensuring that access was securely managed and monitored across their hybrid environment.

Key IAM Migration Considerations

Each of these considerations aligns with the zero trust philosophy of minimizing implicit trust and enforcing least-privilege access. By integrating these practices, organizations create a more resilient IAM framework that actively supports zero trust in hybrid and cloud environments.

1. **Identity Federation and Single Sign-On (SSO):** Implementing identity federation enables users to access multiple services (both cloud and on-premises) with a single set of credentials, enhancing the user experience without compromising security.

2. **Principle of Least Privilege:** Ensuring users and services have only the necessary access to perform their functions minimizes the risk of unauthorized access and potential data breaches.

3. **Automated Provisioning/Deprovisioning:** Automating the creation and removal of user accounts and access rights in cloud services ensures efficient and secure access management, reducing the risk of orphaned accounts or unauthorized access.

4. **Compliance and Regulatory Requirements:** Cloud migration must consider compliance with relevant regulations and standards,

requiring adjustments in IAM policies and practices to meet specific compliance needs.

5. **Continuous Monitoring and Review:** Regularly monitoring IAM policies and access patterns helps detect and respond to anomalies, ensuring ongoing compliance and security in the dynamic cloud environment.

Best Practices and Considerations for IAM Implementation

Implementing multifactor authentication adds an extra layer of security, making it significantly harder for attackers to gain unauthorized access. Regular audits of IAM roles and permissions ensure they align with current security policies and business needs. Also, educating and training staff on IAM best practices and security awareness is crucial to preventing accidental breaches or misuse of resources.

How to Structure Resources and Permissions in the Cloud

Cloud resources are all the tools and applications that make up a cloud computing environment. These include servers, databases, data storage components, and more. Understanding this structure is important for understanding how permissions are assigned and should be managed to maintain security and efficiency in a cloud-native environment. Let’s look at how each cloud provider structures its cloud resources.

As shown in Figure 19-7, each major cloud provider (AWS, Azure, and GCP) offers a unique model for organizing resources and assigning permissions, reflecting different approaches to identity and access management in the cloud.

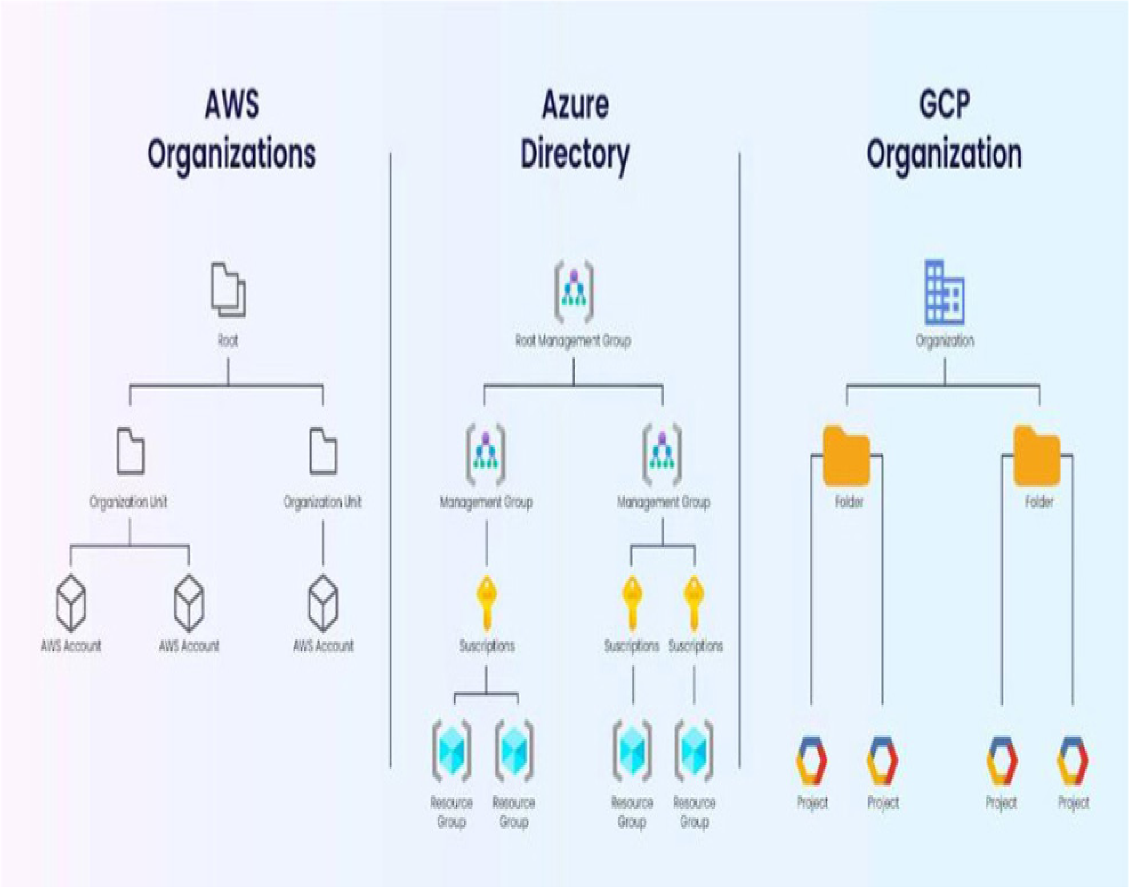


Figure 19-7 *Structure of Resources in AWS, Azure, and GCP*

- **AWS:** This provider uses an account-based model where the main container is an AWS account. Organizations can consolidate multiple accounts using AWS Organizations. This model can use a single account for all application stages or separate accounts for development, staging, and production environments.

- **Azure:** This provider implements a hierarchical RBAC model starting with resource groups that contain resources for a specific application. Subscriptions hold resource groups, and management groups hold subscriptions, culminating in a root management group. This structure is designed to mirror an enterprise’s organizational structure.

- **GCP:** Similar to Azure, GCP’s basic container is a project, which houses resources for a single application. Folders can contain multiple projects, supporting a nested organizational structure. Projects also serve as the billing unit.

Permissions and IAM

In AWS, permissions are specified in an IAM Policy document, combining permissions and resources in a single JSON file. This approach does not separate permissions from resources, contrasting with Azure and GCP, where both utilize a model that decouples permissions from resources. Permissions are assigned at various scopes, and inheritance is enabled, meaning roles at higher scopes have broader permissions. This separation allows for more granular and flexible access control.

AWS Example: Identity-Based Policy for EC2 Access

Example 19-1 shows an example of an identity-based policy that allows full EC2 access within a specific region. This policy defines permissions for programmatic and console access.

Example 19-1 *AWS Example: Grant Full Access to All EC2 Actions Within the `us-east-2` Region*

```
{
   “Version”: “2012-10-17”,
   “Statement”: [
       {
           “Action”: “ec2:*”,
           “Resource”: “*”,
           “Effect”: “Allow”,
           “Condition”: {
                “StringEquals”: {
                    “ec2:Region”: “us-east-2”
                }
            }
       }
   ]
}
```

Azure Example: Role-Based Access Control (RBAC)

In Azure, role-based access control is used to manage permissions at different hierarchical levels such as management groups, subscriptions, resource groups, and individual resources. Example 19-2 shows an example of a custom role that grants full access to a specific resource group.

Example 19-2 *Azure Example: Custom Role Definition*

```
{
 “Name”: “CustomRole”,
 “Id”: “12345678-1234-5678-1234-567812345678”,
 “IsCustom”: true,
 “Description”: “Full access to resource group”,
 “Actions”:[
   “*”
 ],
 “NotActions”:[],
 “DataActions”:[],
 “NotDataActions”:[],
 “AssignableScopes”:[
   “/subscriptions/{subscription-id}/resourceGroups/{resource-gro
 ]
}
```

Here, a custom role is assigned to a user at the resource group level. Replace {subscription-id}, {resource-group-name}, and {user-object-id}, shown in Example 19-3, with the actual values.

Example 19-3 *Azure Example: Role Assignment*

```
{
 “roleDefinitionId”: “/subscriptions/{subscription-
id}/providers/Microsoft.Authorization/roleDefinitions/12345678-12
567812345678”,
 “principalId”: “{user-object-id}”,
 “scope”: “/subscriptions/{subscription-id}/resourceGroups/{resou
}
```

GCP Example: IAM Policy for Compute Engine

In GCP, IAM policies are defined in JSON and attached to resources.

Here’s an example of a policy that grants full access to all compute engine resources within a specific project. Replace {project-id} and example-user@gmail.com, shown in Example 19-4, with the actual project ID and user email.

Example 19-4 *GCP Example: Policy Binding at the Project Level*

```
{
 “bindings”: [
   {
     “role”: “roles/compute.admin”,
     “members”: [
       “user:example-user@gmail.com”
     ]
   }
 ],
 “resource”: “projects/{project-id}”
}
```

To add hierarchical levels, you would structure it with folders and projects. To do so, replace {folder-id} in Example 19-5 with the actual folder ID.

Example 19-5 *GCP Example: Policy Binding at the Folder Level*

```
{
 “bindings”: [
   {
     “role”: “roles/compute.admin”,
     “members”: [
       “user:example-user@gmail.com”
     ]
   }
 ],
 “resource”: “folders/{folder-id}”
}
```

Explanation and Context

Each cloud provider offers tools for managing user access management, such as IAM users in AWS, Azure Active Directory in Azure, and Google Cloud Identity or Google Workspace in GCP. The management of service identities also varies (service access management), with AWS and GCP using roles and service accounts, respectively, and Azure employing managed identities and service principals, for resource access, eliminating the need to manage credentials manually. Figure 19-8 details the different constructs and the relation among them for each of the three hyperscalers.

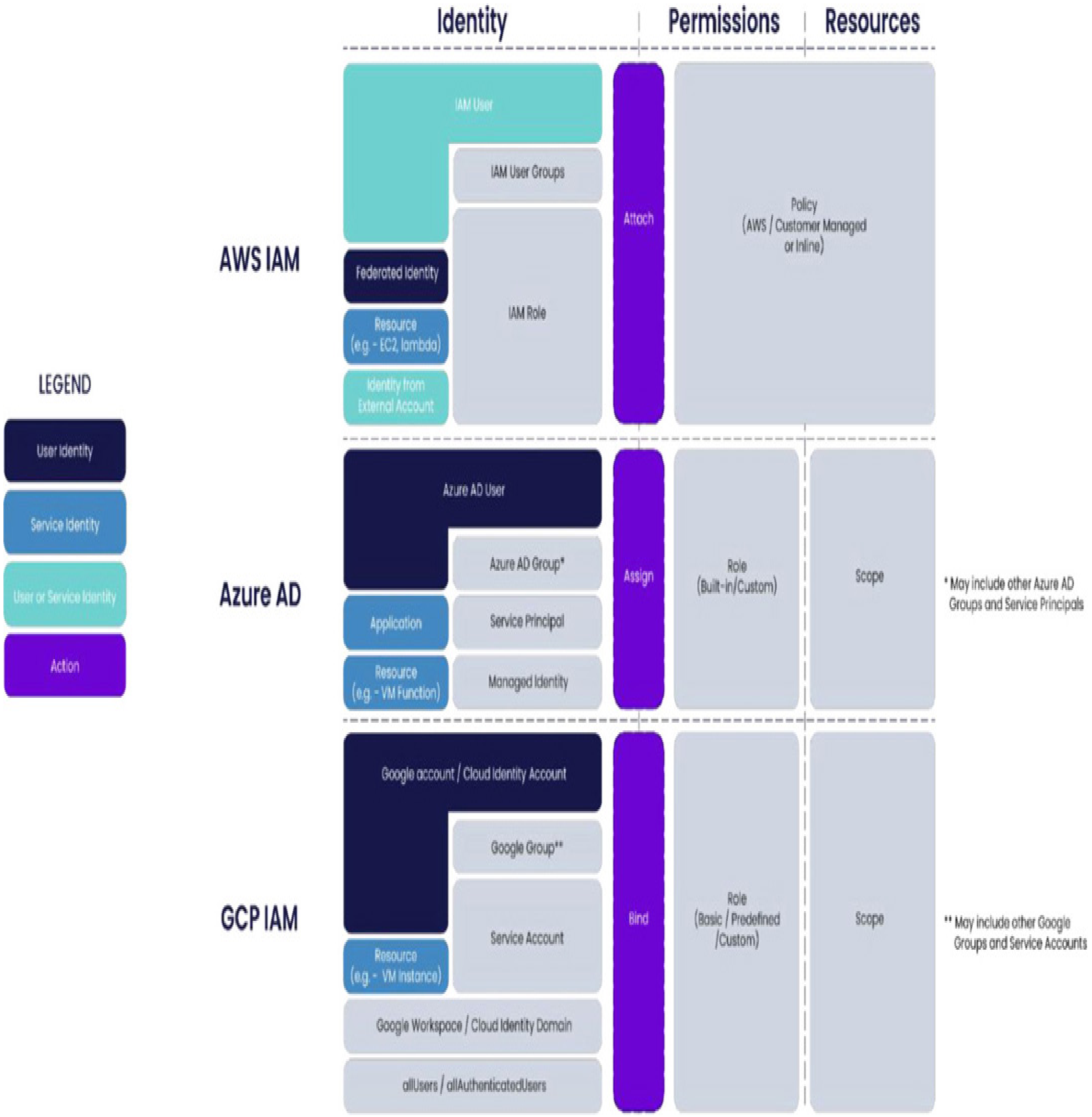

Identity
Permissions
Resources
AWS IAM
IAM User
IAM User Groups
Federated Identity
Resource (e.g. - EC2, lambda)
Identity from External Account
IAM Role
Attach
Policy (AWS / Customer Managed or Inline)
LEGEND
User Identity
Service Identity
User or Service Identity
Action
Azure AD
Azure AD User
Azure AD Group*
Application
Service Principal
Resource (e.g. - VM Function)
Managed Identity
Assign
Role (Built-in/Custom)
Scope
* May include other Azure AD Groups and Service Principals
GCP IAM
Google account / Cloud Identity Account
Google Group**
Service Account
Resource (e.g. - VM Instance)
Google Workspace / Cloud Identity Domain
allUsers / allAuthenticatedUsers
Bind
Role (Basic / Predefined /Custom)
Scope
** May include other Google Groups and Service Accounts

Figure 19-8 *Permissions Listing in AWS, Azure, and GCP*

- **AWS:** AWS uses an account-based model where the main container is an AWS account. Permissions are managed through IAM policies, which can be identity-based or resource-based. While AWS does allow for some hierarchical organization through AWS Organizations, it primarily operates on an account level rather than a deeply nested hierarchy.

- **Azure:** Azure implements a hierarchical model starting with resource groups within subscriptions and management groups. This structure mirrors an enterprise's organizational hierarchy and allows for granular control at various levels.

- **GCP:** GCP uses a project-based model where the main container is a project. Projects can be nested within folders, and permissions are inherited from higher levels down to individual resources. This allows for flexible and hierarchical access control similar to Azure.

External Access and Security Considerations

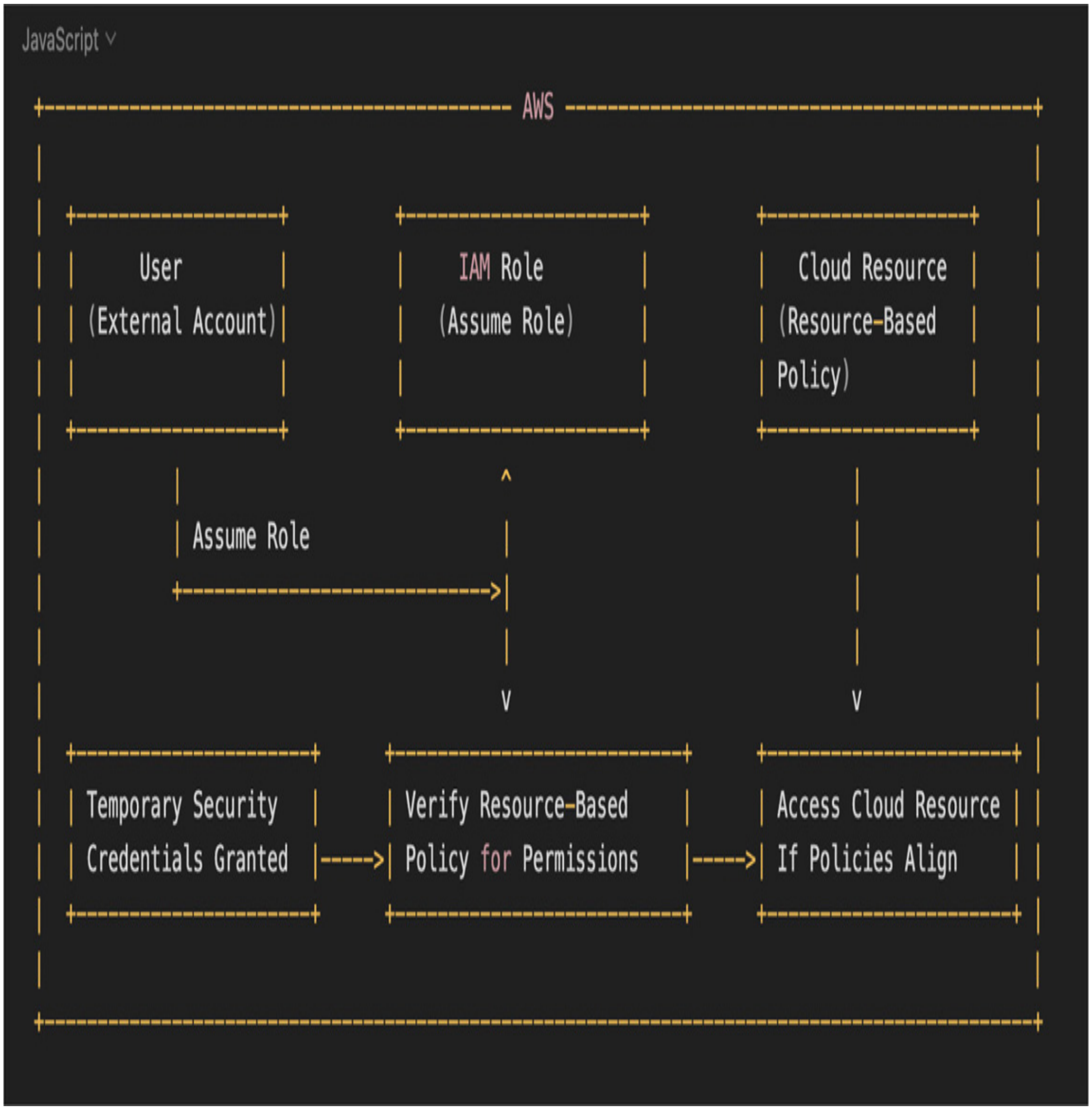
In many cases, organizations have to provide third-party access, such as getting some support services from a third party. Also, here are some common security pitfalls to avoid: static credentials, blind trust on policies managed by provider, and accidental public access configurations. The figures in the following sections illustrate how external access is provided to third parties in AWS, Azure, and GCP, focusing on the mechanisms unique to each cloud provider.

AWS

In AWS, a resource-based policy allows external access. Access to an identity has a resource-based policy that describes permissions, which is attached to the resource instead of the identity, because the resources are managed by administrators, not identities. An admin cannot create a resource-based policy for each resource for which access is required, because it is not scalable, and so instead can create an external IAM role for assumption. This type of policy specifies which external accounts are granted access. This IAM role is used by external users to access the cloud resource.

AWS External Access Workflow: IAM Role with Resource-Based Policy:

Figure 19-9 represents how AWS provides external access using IAM roles and resource-based policies.



```
+-------------------------------------- AWS --------------------------------------+
|                                                                                 |
|  +-----------------+        +--------------------+        +-----------------+   |
|  |      User       |        |      IAM Role      |        |  Cloud Resource |   |
|  | (External Account)|      |    (Assume Role)   |        | (Resource-Based |   |
|  |                 |        |                    |        | Policy)         |   |
|  +-----------------+        +--------------------+        +-----------------+   |
|           |                          ^                            |             |
|           | Assume Role              |                            |             |
|           +------------------------->|                            |             |
|                                      |                            |             |
|                                      v                            v             |
|  +-------------------+    +-----------------------+    +---------------------+  |
|  | Temporary Security |   | Verify Resource-Based |    | Access Cloud Resource | |
|  | Credentials Granted |---->| Policy for Permissions |---->| If Policies Align | |
|  +-------------------+    +-----------------------+    +---------------------+  |
|                                                                                 |
+---------------------------------------------------------------------------------+
```

Figure 19-9 *AWS Example: IAM Role with Resource-Based Policy*

- **Key Elements:**
  - **User:** This represents the external account or identity requesting access to AWS resources.
  - **IAM Role:** This temporary identity in AWS is assumed by the external user, containing the permissions required to access specific resources.
  - **Cloud Resource:** This is the AWS resource to be accessed, such as an S3 bucket, EC2 instance, or Lambda function.

  - **Resource-Based Policy:** This policy is attached directly to the resource, specifying which external accounts or roles can access it.

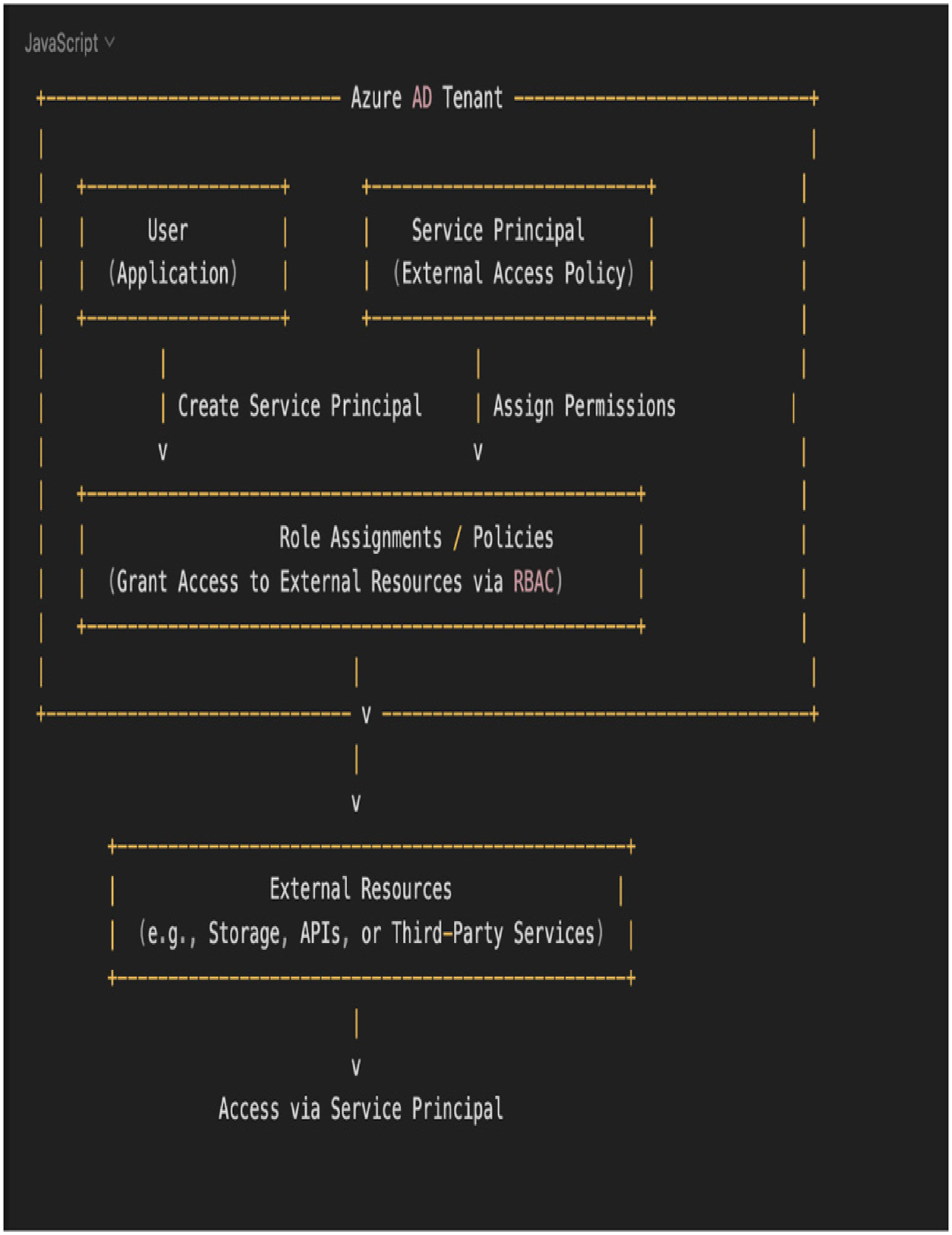
- **AWS External Access Workflow: IAM Role with Resource-Based Policy:**
- The external user or account assumes an IAM role using an assume role policy.
- The IAM role contains the permissions required to access the AWS resource.
- The cloud resource enforces access control using a resource-based policy, specifying who can access it.
- Access to the resource is granted if both the IAM role and resource-based policy align.

Azure

In Azure, keeping the service principal separate from the app allows you to give the application external access against other tenants. A service principal is created when the app is actually installed in a tenant to allow external access. That service principal is the application with permissions to external resources in other tenants. This model enables applications to obtain permissions to access resources across tenants, thereby providing secure and seamless external access.

Azure External Access Workflow: Service Principal with External Tenant

Figure 19-10 illustrates how Azure provides external access using service principals.



```
+--------------------------- Azure AD Tenant ---------------------------+
|                                                                       |
|   +-----------------+      +--------------------------+              |
|   |      User       |      |    Service Principal     |              |
|   |  (Application)  |      | (External Access Policy) |              |
|   +-----------------+      +--------------------------+              |
|           |                          |                               |
|           | Create Service Principal | Assign Permissions           |
|           v                          v                               |
|   +--------------------------------------------------+              |
|   |           Role Assignments / Policies            |              |
|   |  (Grant Access to External Resources via RBAC)   |              |
|   +--------------------------------------------------+              |
|                            |                                          |
+---------------------------- v ----------------------------------------+
                             |
                             v
      +------------------------------------------------+
      |               External Resources               |
      |  (e.g., Storage, APIs, or Third-Party Services) |
      +------------------------------------------------+
                             |
                             v
                 Access via Service Principal
```

Figure 19-10 *Azure Example: Service Principal with External Tenant*

- **Key Elements:**
  - **User:** This represents the external application or identity requesting access to Azure resources.

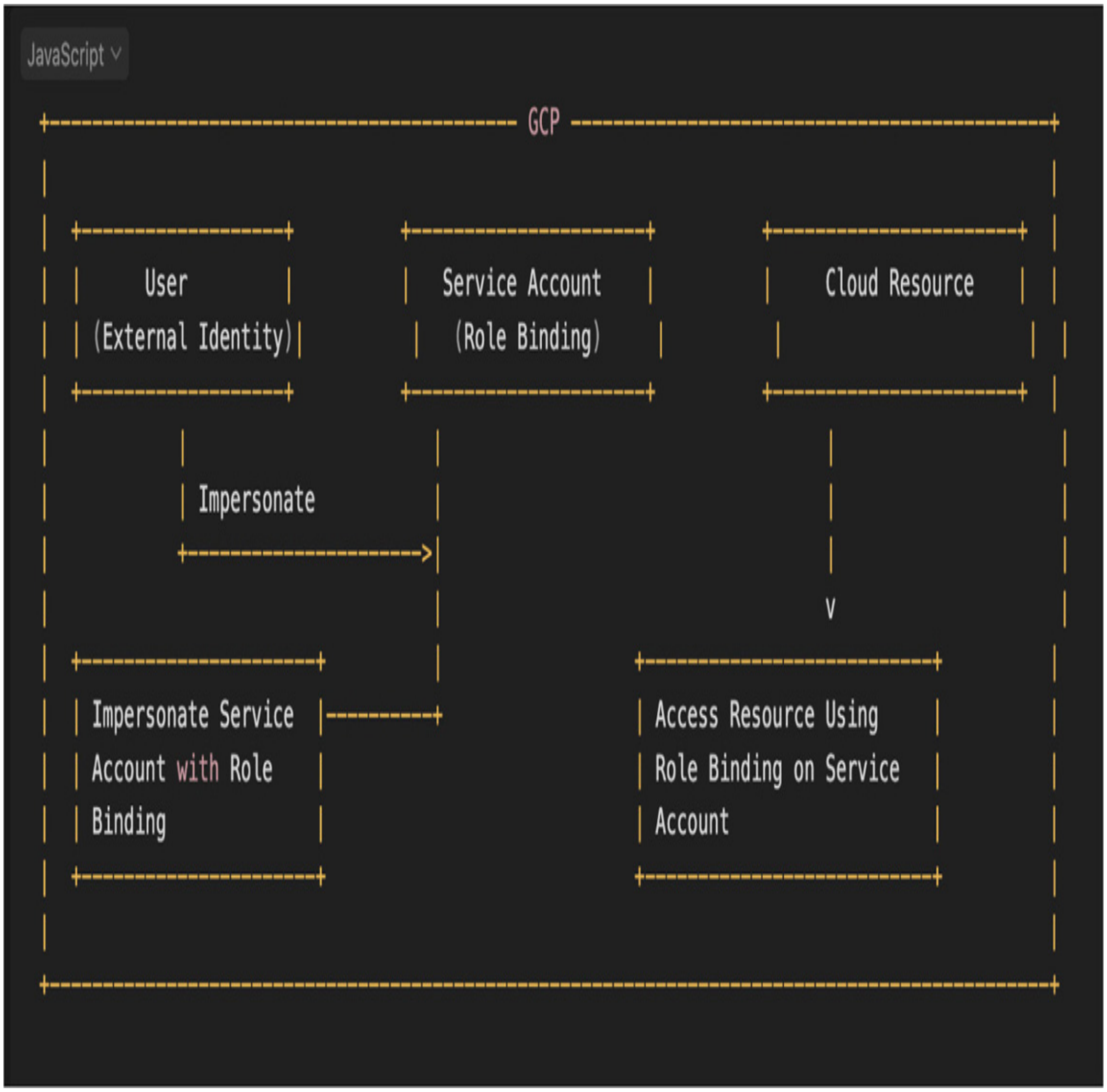
    - **Service Principal:** This security identity created in Azure AD acts on behalf of the external application, enabling secure access to resources.
    - **Cloud Resource:** This is the Azure resource to be accessed, such as a virtual machine, Azure Storage, or key vault.
    - **External Access Policy:** This policy defines permissions for the service principal, granting access to specific Azure resources.
- **Azure External Access Workflow: Service Principal with External Tenant:**
- The external application creates a service principal in Azure Active Directory for access.
- The service principal is granted permissions through role assignments or an external access policy.
- The service principal credentials are used by the application to authenticate and access Azure resources securely.
- Access is restricted and managed based on permissions configured for the service principal.

GCP

In GCP, a service account is used to provide access to third parties. GCP provides access by simply creating a binding for an external identity on a service account with a role that can be impersonated. The external identity impersonates the service account, which has the required role to access the cloud resources. This method provides a straightforward way to manage external access by binding roles to service accounts that can be securely impersonated by external users.

GCP External Access Workflow: Service Account with Role Binding

Figure 19-11 shows how GCP provides external access using service accounts and role bindings.



```
+-------------------------------------- GCP --------------------------------------+
|                                                                                 |
|  +-----------------+        +-------------------+        +-------------------+  |
|  |      User       |        |  Service Account  |        |  Cloud Resource   |  |
|  |(External Identity)|       |   (Role Binding)  |        |                   |  |
|  +-----------------+        +-------------------+        +-------------------+  |
|            |                  |                               |                |
|            | Impersonate      |                               |                |
|            +----------------->|                               |                |
|                               |                               v                |
|  +-------------------+        |              +----------------------+          |
|  | Impersonate Service |---------+            | Access Resource Using |         |
|  | Account with Role |                        | Role Binding on Service |       |
|  | Binding           |                        | Account               |         |
|  +-------------------+                        +----------------------+          |
|                                                                                 |
+---------------------------------------------------------------------------------+
```

Figure 19-11 *GCP Example: Service Account with Role Binding*

- **Key Elements:**
  - **User:** This represents the external identity or third-party entity requesting access to GCP resources.
  - **Service Account:** This Google Cloud identity acts on behalf of the external user, holding the permissions needed to access resources.
  - **Role Binding:** This binding associates the service account with a predefined role that grants specific permissions to GCP resources.
  - **Cloud Resource:** This is the GCP resource to be accessed, such as a compute engine instance, cloud storage bucket, or BigQuery

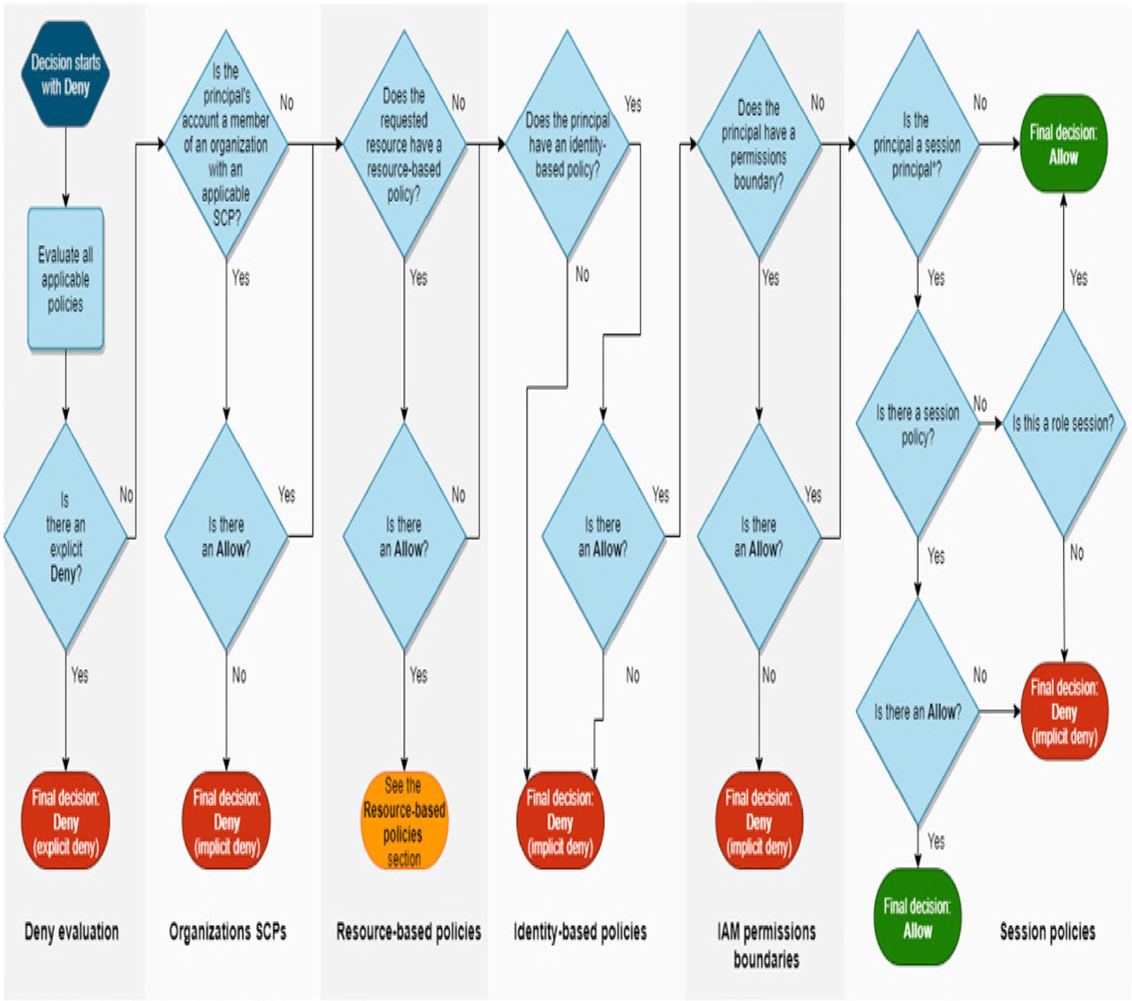
dataset.

- **GCP External Access Workflow: Service Account with Role Binding:**
- The external identity impersonates a service account created in GCP.
- The service account is assigned a role binding, granting specific permissions.
- The service account uses its permissions to access the cloud resource.
- The cloud resource enforces access control based on the permissions defined in the role binding.

Guardrails and Automated Analysis

Cloud providers offer various guardrails to help manage access and enforce security policies. Examples include AWS's multistage evaluation process, GCP's IAM Deny policies, and Azure's resource locks and deny assignments. Let's evaluate AWS IAM as an example for determining whether a request is allowed or denied within an account.

The flow chart in Figure 19-12 provides details about how the decision is made.


Decision starts with Deny
Evaluate all applicable policies
Is there an explicit Deny?
No
Yes
Final decision: Deny (explicit deny)
Deny evaluation
Is the principal's account a member of an organization with an applicable SCP?
No
Yes
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
Organizations SCPs
Does the requested resource have a resource-based policy?
No
Yes
Is there an Allow?
No
Yes
See the Resource-based policies section
Resource-based policies
Does the principal have an identity-based policy?
Yes
No
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
Identity-based policies
Does the principal have a permissions boundary?
No
Yes
Is there an Allow?
Yes
No
Final decision: Deny (implicit deny)
IAM permissions boundaries
Is the principal a session principal*?
No
Yes
Final decision: Allow
Is there a session policy?
No
Yes
Is this a role session?
Yes
No
Is there an Allow?
No
Yes
Final decision: Deny (implicit deny)
Final decision: Allow
Session policies

*A session principal is either a role session or an IAM federated user session.

Figure 19-12 *AWS Policy Evaluation Flow Chart*

AWS Policy Evaluation Logic Summary and Flow Chart:

1. **Implicit Deny by Default:**
   - By default, all requests are denied (implicit deny), except for the AWS account root user, which has full access.
2. **Explicit Deny:**
   - If any policy contains an explicit deny, the request is denied. This is evaluated first to quickly resolve denials.
3. **Organization's SCPs:**
   - SCPs are evaluated next. If no Allow is found in SCPs, the request is denied.

- If SCPs allow the request, evaluation continues.

4. **Resource-Based Policies:**
   - These policies are evaluated if applicable, especially for IAM roles, IAM users, and session principals.
   - An Allow in a resource-based policy can grant access even if other policies have implicit denies.
5. **Identity-Based Policies:**
   - These policies are checked for the requesting principal.
   - If no Allow is found in the identity-based policies, the request is implicitly denied.
   - If an Allow is found, evaluation continues.
6. **IAM Permissions Boundaries:**
   - These boundaries are checked for the IAM entity.
   - If the boundary does not allow the action, the request is denied.
   - If allowed, evaluation continues.
7. **Session Policies:**
   - These policies are checked for session principals (IAM role sessions or federated user sessions).
   - If the session policy does not allow the action, the request is denied.
   - If allowed, evaluation continues.
8. **Final Decision:**
   - If all applicable policies (identity-based, resource-based, SCPs, permissions boundaries, session policies) allow the action, the request is allowed.
   - Any explicit deny results in a final decision of Deny.

Special Cases:

- **IAM Role Session:** Evaluated separately. An explicit allow in a resource-based policy for the role session ARN can override implicit denies.
- **Root User:** Always allowed unless explicitly denied by SCPs.
- **AWS Service Principals:** Allowed if explicitly permitted in resource-based policies.

By following this evaluation logic, AWS ensures that all requests are thoroughly checked against various policies to maintain security and compliance within the account.

The impact of resource-based policies and implicit denies in other types of policies is outside the scope of this book, but you can view AWS's IAM User Guide: Policy Evaluation Logic (see the "References" section).

Automated Analysis Solutions

An automated analysis platform can further enhance security by continuously analyzing permissions, identifying excessive or unnecessary access, and recommending remediations to achieve the principle of least privilege.

Automated analysis solutions play a crucial role in cloud security, offering the ability to continuously monitor, analyze, and manage the security posture of cloud environments. Both hyperscalers (like AWS, Azure, and GCP) and specialized security companies (such as Cisco and others) offer solutions designed to enhance visibility, enforce compliance, and ensure the principle of least-privilege across cloud infrastructures.

These solutions demonstrate how automated analysis can significantly enhance an organization's ability to manage cloud security effectively, by providing continuous monitoring, detailed insights, and actionable recommendations:

Following are some examples of hyperscaler services:

- **AWS Security Hub:** This service provides a comprehensive view of your security alerts and security posture across your AWS accounts. It aggregates, organizes, and prioritizes security findings from various AWS services, such as Amazon GuardDuty, Amazon Inspector, and AWS IAM Access Analyzer, as well as from AWS Partner Network (APN) security solutions. It helps automate security checks and manage security standards, including the Center for Internet Security (CIS) AWS Foundations Benchmark.

- **Azure Security Center:** This service (recently renamed to Microsoft Defender for Cloud) offers unified security management and advanced threat protection across hybrid cloud workloads. It automatically assesses and recommends improvements to your cloud resources' security configurations, identifies and helps remediate vulnerabilities, and provides advanced threat protection and detection capabilities. It also integrates with various Azure services to enhance visibility and control over your security posture.
- **Google Cloud Security Command Center (Security Command Center):** This service helps identify and organize security findings in your Google Cloud resources. It provides visibility into cloud assets, detects misconfigurations and vulnerabilities, and provides insights to reduce exposure to threats. With integrated tools like the Event Threat Detection, Cloud DLP (Data Loss Prevention), and Web Security Scanner, Security Command Center enables comprehensive security and risk management.
- **Cisco Attack Surface Management:** This service, previously known as Cisco Secure Cloud Insights, is Cisco's cloud-native cybersecurity platform designed to provide visibility, compliance, and protection against threats for cloud applications and data. By leveraging APIs, Cisco Attack Surface Management tracks activities across the cloud environment, detects anomalies, enforces security policies, and offers real-time threat protection. The platform operates across multiple cloud environments, offering comprehensive insights into user behavior, sensitive data, and application security, thereby aiding organizations in enforcing compliance standards and mitigating risks.

To defend against increasingly sophisticated digital threats, organizations require comprehensive network visibility, total control, and rapid response capabilities. Cisco Attack Surface Management identifies vulnerabilities in near real-time, enabling swift response actions. This solution provides a thorough examination of the attack surface through relationship mapping, reduces risks, and ensures compliance with 100 predefined API integrations. It allows for effective scrutiny of the entire environment, maintaining compliance and security posture by monitoring relationships across hybrid IT and multiple cloud environments. With over 800 premade queries or customizable options, it identifies security gaps and misconfigurations. As part of Cisco XDR, Cisco Attack Surface Management offers a holistic inventory of all entities and current security risks, aiding in identifying and remediating misconfigured cloud environments and asset vulnerabilities.

By combining the situational awareness from Secure Cloud Analytics with the structural insights of Cisco Attack Surface Management, organizations can enhance the security of their digital environments. This integration provides visibility into vulnerabilities and anomaly-based threat vectors, further strengthening the overall security posture.

Summary

In this chapter, we covered the key characteristics related to workload mobility with a focus on moving applications, services, and data from on-premises infrastructure to the cloud in a secure and efficient manner. We explored the benefits of using the cloud, such as scalability, affordability, and ease-of-use for disaster recovery, along with the challenges of maintaining data security, compliance, and application compatibility. Based on real-world examples such as ABC Corp, this chapter revealed phased migration strategies, the Seven R’s framework, and the criticality of aligning application migration activities with the core tenets of zero trust. The chapter also presented practical tools available from the cloud providers, as well as the recommended processes of transferring data securely end-to-end, what hybrid IT model technologies best suit organizations, and how to seek the optimal position for an organization’s workload, providing a full view of the cloud journey.

Organizations need to start with a straightforward migration strategy aligned with business objectives, with the right workload migration strategy in mind, prioritizing workloads based on criticality and adopting phased approaches to minimize business disruption for a successful and secure transition of workloads to the cloud. Zero trust concepts like the principle of least-privilege access and strong identity management should be embedded throughout to provide reinforced protection. Encryption of data in transit and at rest and secure transfer methods are all essential for protecting data from unwanted access. Organizations need to replace legacy systems for paradigm and scalability and introduce teams to new competencies through training to prepare them for effective strategy execution. Once the migration process is complete, however, it's essential to prioritize performance evaluation, cost management, and optimization to ensure that your organization can harness the long-term value of adopting cloud-based solutions.

References

1. "2024 State of the Cloud Report": https://info.flexera.com/CM-REPORT-State-of-the-Cloud-2024-Thanks
2. "7 AWS Strategies for Migrating to the Cloud": https://aws.amazon.com/blogs/enterprise-strategy/new-possibilities-seven-strategies-to-accelerate-your-application-migration-to-aws/
3. "AWS, Azure and GCP: The Ultimate IAM Comparison": https://www.tenable.com/blog/aws-azure-and-gcp-the-ultimate-iam-comparison
4. "Policy Evaluation Logic": https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_evaluation-logic.xhtml
5. "How End-to-End Visibility Can Help with IT Migration": https://news-blogs.cisco.com/emea/2023/06/05/how-end-to-end-visibility-can-help-with-it-migration/
6. "Building a Cost-Effective Full Observability Solution Around Open APIs and CNCF Projects":

https://www.chaossearch.io/blog/building-cost-effective-observability-solution-w-open-apis-cncf

7. Cisco GitHub repository for the workshop: https://github.com/cisco-emea-cx-cto/CLUS24-Workshop-LTRCLD-1370-Accelerating-your-Cloud-Native-Observability-via-Open-Telemetry

8. “Converging Infrastructure Monitoring and Observability Using OpenTelemetry”: https://arielleza.notion.site/cl24-ltrcld-1370-part2-converging-infra-monitoring-and-otel

Part 5: Key Customer Case Studies

Chapter 20. Resilience and Survivability

In this chapter, you will learn about the following:

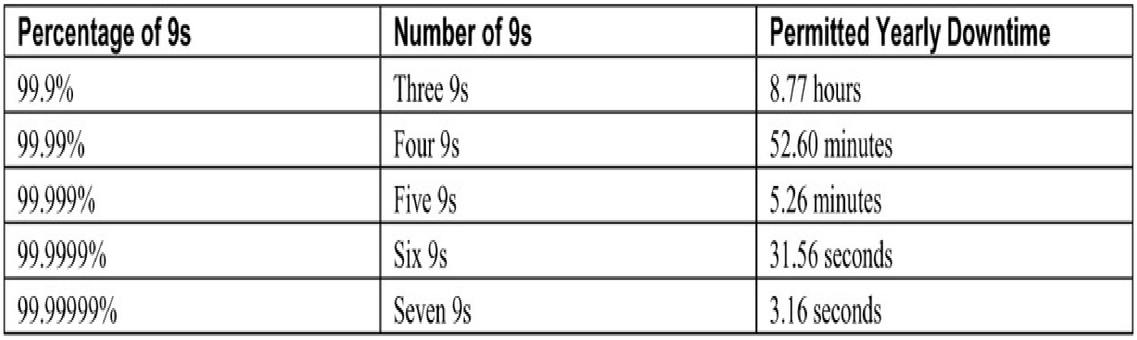
- Resilience metrics
- Types of resilience
- Software resilience strategies
- Resilience in the cloud
- Consequences of AAA resilience
- Audit trail redundancy
- Chaos engineering

Resilience Metrics

A key aspect of any architecture that is associated with critical infrastructure that needs appropriate planning, in conjunction with the security and compliance aspects of the deployment, is the actual resilience of the end-to-end architecture that is deployed. Deploying software-defined network architectures comes with numerous aspects that require evaluation. This evaluation needs to consider the consequences of taking shortcuts when planning redundancy, resilience, and survivability.

Deciding how redundant an architecture and its corresponding services should be is not always an easy task, and when we're talking with nontechnical stakeholders, they often raise the lofty request of achieving five 9s (99.999 percent) uptime, as shown in Table 20-1, or similar levels of availability. While, on paper, this can sound like a commendable ask, clearly for business stakeholders who are involved in making corporate decisions, it sounds as though it could prevent them from being blamed for a potential period of services or systems being unavailable. In reality, such an ask tends to be unrealistic and can result in significant operation costs in terms of resources, planning, residual systems, test architectures, and global coordination, just to achieve that extra 0.001 percent.

Table 20-1 *Resilience Metrics Percentage of Uptime vs. Time Calculated over a One-Year Period*



Percentage of 9s	Number of 9s	Permitted Yearly Downtime
99.9%	Three 9s	8.77 hours
99.99%	Four 9s	52.60 minutes
99.999%	Five 9s	5.26 minutes
99.9999%	Six 9s	31.56 seconds
99.99999%	Seven 9s	3.16 seconds

The process of identifying the resilience needed within an architecture should be rooted in the per-scenario needs of the company, and this level of uptime can vary greatly within the same organization based on function and service. For instance, does the wireless network in a hospital that is being used for both guest network access and vital health monitoring equipment for patients require the same level of resilience in the parking garage as it does in patients' rooms where vital systems are located? Do application systems that serve a particular language or dialect need to operate with four or five 9s resilience outside of operating hours (for instance, between 00:00 and 06:00), or is it acceptable to perform maintenance activities on these systems during those hours? If a critical security patch or update is released, within what period of time will patch testing be concluded? How quickly should the patch rollout take place? And how quickly should that rollout be concluded globally? These are all important questions to ask and only provide a starting point. Such questions should serve as the groundwork toward defining the right service-level objectives and service-level indicators, as well as support the combined negotiation and agreements for what would be possible to commit to in the form of service-level agreements (SLAs) for an organization.

Variance of availability by sector and industry has a direct correlation with downtime for the corresponding services that the business may provide. For instance, in the domain of web hosting, the general availability metrics that are considered acceptable are in the ballpark of 99.9 percent availability. In contrast, availability in e-commerce and banking systems has a much more tangible impact in terms of financial loss, so there tends to be an expectation of 99.99 percent availability. Looking toward even more critical domains such as healthcare, where vital monitoring systems may be connected to the IT infrastructure, a loss in communication can literally be the difference between life and death, with availability expectations being targeted toward 99.999 percent. While some scenarios demand even more aggressive availability values, they tend to be limited to architectures and services related to wartime or military activity.

A key question you need to ask when planning, designing, and building out an architecture in the context of security and zero trust is "What would happen if ?"

- What would happen if I lose my single uplink to my ISP?
- What would happen if the fan or power supply fails on one of my routers?
- What would happen and what impact would be caused if the entire router fails?
- What would happen if I lose my connection to my TACACS+ or RADIUS infrastructure?
- What would happen if the firewalls to a critical data center become overloaded or unstable?
- What would happen if my SaaS service for MFA becomes unreachable?
- What would happen if my SMS gateway provider encounters a disruption?

These examples of scenarios that could happen in any IT architecture may be trivial, but each example means different things based on the core business of the organization and what such disruptions would effectively mean.

While the dream of having services with an uptime of five 9s or the lofty and often unrealistic aspiration of having a service with 100 percent uptime sounds good in theory, in practice, the balance of cost, operational agility, and security needs to be put under a microscope to identify the best approach for the organization. Also, note that providing an extremely high level of resilience for various services can also lead to unexpected and sometimes catastrophic consequences. For example, some companies get into the bad habit of using public IP addresses for common free Internet services such as DNS, as a means to verify network reachability for their services. In that scenario, if the central service is not available, failovers could take place within data centers because the operators falsely conclude that the Internet is no longer reachable and there must be an Internet service provider (ISP) problem. Unfortunately for the operators who decide to build their service verification based on this single factor, when the central service goes down for routine maintenance, the result is a global outage because all of the services are built on the same incorrect assumption that the dependent system is always reachable.

The situation just described is an extreme example of a mistake made as part of the resilience logic that an organization (or, in reality, a single operator) takes for a particular critical service, but it needs careful consideration as a whole when planning services, security, and survivability. If you consider how many of the hyperscalers and other organizations that drive services through site reliability teams manage their networks and service-level objectives (SLOs), they are cautious to ensure that the uptime of a service or system does not exceed its respective target. This approach prevents teams from becoming dependent on the service always being available and often prevents incorrect conclusions from developer and application teams that have this expectation too when developing their architectures to interface with external services or services that are being developed by other teams.

What does any of this have to do with security? As modern IT architectures and networks grow, a tighter coupling is taking place between the classic infrastructure, data centers, firewalls, security appliance, and observability tools into the data and core business application domains. This transition sees the network that, that for many application owners and business teams, was considered to be "the plumbing", namely a means to get to the Internet. To a role beyond only being an enabler to access a specific website or service, instead becoming a key component in the end-to-end provisioning flows that enable new offers to come to market more rapidly, and an information source of intelligence and insights that may exist in the data including confirmation of nonrepudiation within a given dataset. Clive Humby, a UK-based mathematician, coined the phrase “data is the new oil,” which in recent times has been paraphrased as “data is the new gold.” The reason for this perspective is that through methods of predictive analytics against larger and broader data sets, which include network-based identifiers such as source location, many organizations come to conclusions about their core businesses that allow the right directions to steer their companies. In scenarios where architectural resilience and security may not be properly deployed, data corruption, data tampering, and sometimes data loss could occur. These consequences could result in the data set that is stored not being useful or representative enough for its intended purpose, precluding use for data science activities.

Identifying the right balance of security and operational agility is critical for every organization today because the levels of threats that organizations are encountering and having to deal with are perpetually increasing. In addition, some industry voices and analysts are predicting that AI-based agents will represent 20 percent of all threat actors by 2027. As many network and application breaches take place, through a combination of multiple vulnerabilities operating simultaneously, resilience strategies are becoming ever more focused on what can be done to avoid putting all your efforts into only one solution. Which approach should be taken to ensure that if one line of defense fails or needs to be bypassed due to software- or hardware-related issues, what further layers of security are in place to ensure that sustained operations are possible? In addition to these layers, is the right strategy in place to identify when and how the architecture was breached, and are the right tools available to lock down the service rapidly and dynamically to avoid impact?

Many organizations pursue a multivendor strategy. This approach is not uncommon and is quite often considered a business continuity decision in organizations such as service providers. There is seldom a scenario where a particular vendor is perfect in all categories, and usually, operating an architecture with a single vendor may result in certain trade-offs in terms of feature and functionalities. It is often true that vendors have their own “special sauce” (unique differentiator) and provide variations in functionalities and security features that may be named similarly. A clear understanding of the end-to-end interoperability of a solution is important, because as great as a product from a specific vendor may be in terms of features, how does it operate in terms of Day 2 operations? Does the system leverage working and functional application programming interfaces (APIs) that can integrate with other dependent systems, and are the requisite metrics, traces, and logs available in a manner that can be integrated into external security information and event management (SIEM) or extended detection and response (XDR) architectures for common and centralized consumption? Also, note that if a multivendor strategy is being pursued for a particular architecture—for instance, the firewalls within an organization—how is the delta in the variation between security features satisfied? Does the organization pursue a strategy of deploying a minimal capability, ensuring that nothing would be used that is not possible on both vendor platforms? Or does it pursue an approach that results in entropy between vendor functionalities and variation in the overall operational landscape of the estate?

Types of Resilience

Resiliency is often described as the ability for a system or service to quickly recover from disruptions and continue operating. Resilience factors in multivendor architectures need to be considered in a number of areas, as we will discuss in the following sections.

Physical Resilience

Physical resilience is the deployment of equipment that could represent a warm, cold, or tertiary backup within an architecture. This could range from backup firewalls that are installed in another physical server room within the same building, or it could also represent a full separation of the redundant components to different buildings, states, cities, or in some cases, even different countries or continents.

Quite often physical resilience is considered, but the question posed in the form of a doomsday scenario is seldom pondered at length. Often resilience is considered through physical redundancy, but only in the context of a single component potentially failing and the respective repercussions associated with that failure. More significant concerns, such as a building collapse, fire, flood or landslide, seldom are considered.

While the expression “never say never” is sometimes raised when considering survivability and resilience, a clear business risk evaluation and assessment are not often the focus when technology teams make decisions that are relevant to the placement of equipment. Under certain circumstances, this placement can end in catastrophe when the right planning is not in place.

One example where such planning went terribly wrong took place around 10 years ago (around 2015) with a service provider that was responsible for providing Internet, cellular, and fixed-line telephony services. The service provider had a strong market position and had a network containing components from all the top global vendors at the time. In addition to new components representing the provider’s Internet estate, many points of presence (PoPs) also were housed with components that were responsible for the telephony network that had been in use for many decades. The general attitude toward this highly reliable equipment was “If it ain’t broke, don’t fix it.” This approach had served the service provider well in the past, and nobody within the organization really paid further attention to the equipment or maintained ongoing communications with the vendors that were responsible for it after installation and purchase. In this particular point of presence, an old analog telephony component landed in a marginal state, shutting off its cooling fan, instead of shutting down like it was supposed to. The device overheated and generated a fire in the facility, quickly engulfing the whole room in flames, destroying and melting all the components in the room. Given the age of some of the plastics used in the old equipment and the toxic gases that they released during the fire, entry into the room was forbidden until a complete chemical cleanup of the room could be performed.

Thankfully, the fire didn’t result in any injury, but the same cannot be said for the redundancy of the location. As is the case for most service providers, a backup location could be activated to ensure that Internet and voice calls would continue, but that architecture was oversubscribed and never intended for long-term use. The carrier began to contact all its vendors to procure replacement equipment. For all the new components, the replacements took place rapidly, allowing the provider to build a temporary PoP for some of its services. One challenge, however, was that the voice telephony equipment was no longer for sale and the vendor that sold it had long gone out of business. Given how deeply integrated that equipment was to technology deployment for that service provider, simply replacing the equipment with new gear was not an option and not a problem that anybody considered from a business continuity perspective.

In the end, the carrier eventually did manage to procure the hardware needed to rebuild the PoP, but given the situation, the provider was relegated to procurement anywhere and everywhere that it could possibly look. In some circumstances, the gear, which was 30 years old, was being sold for a markup of 1000 percent of its market rate on eBay. Unfortunately, that is a price the service provider had to pay to bring its business and resilience back.

This example is a worst-case scenario clearly, and not something that is encountered regularly by any means. But it should provide a reminder that, even with the best-laid plans, operational and security resilience can be impacted when the whole picture is not considered.

Environmental Redundancy

Building out resilience in the context of heating, cooling, and redundant power sources is an important aspect of the deployment of modern data centers and architectures to support high-capacity networks or data infrastructure. With more uncertainty around the maximum and minimum temperatures to expect in different regions, organizations may need to anticipate periods of extended power usage or extended cooling. While workload elasticity can provide some support for moving applications away from a certain location if necessary, this is clearly not a possibility for network infrastructure–related tasks that are directly client serving.

Redundant power connectivity to a building often needs the right level of consideration when contingency planning is concerned, including the use of different entry points for the respective power feeds to the building and backup power or generator usage for critical locations and areas.

Domain Resilience

Domain resilience is a particular area that is fraught with many complexities. A domain in this context is a specific technology grouping; for instance, it can be switching infrastructure, routing infrastructure, firewalls, supplicant security software, an SaaS service, or even a hyperscaler used for a given task.

Is there a requirement or need to build out resilience within domains by deploying variations of models of equipment from the same vendor or even to consider a multivendor deployment strategy?

There is no right or wrong answer here, but organizations need to take this consideration into account when dealing with the increased complexity that comes with entropy within a technical estate. This complexity manifests itself around technical differences in code or software versions, including upgrade procedures that need to be carried out by operations teams in response to new bugs or exploits that are identified. It also includes training and certification track requirements for staff and the complex frameworks of contracts associated with vendor licensing, services, and support for hardware replacement SLAs.

When an organization is considering an in-domain resilience approach, inter-vendor compatibility sometimes comes into play. While things do work in many scenarios, if a different interpretation is taken between one vendor and another in the context of how standards are defined or drafted for protocols used, it can sometimes be a long and drawn-out period of time until arbitration between the two parties finally takes place.

Vendor Resilience

Historically, service providers and militaries have pursued a multivendor strategy. The strategy they generally pursue has a primary and secondary resilience path throughout the architecture, as shown in Figure 20-1. This approach allows for a complete shutdown of a particular vendor's equipment at any point in time, thereby supporting its temporary removal in the case of a critical software defect or limitation that would represent a severe breach.

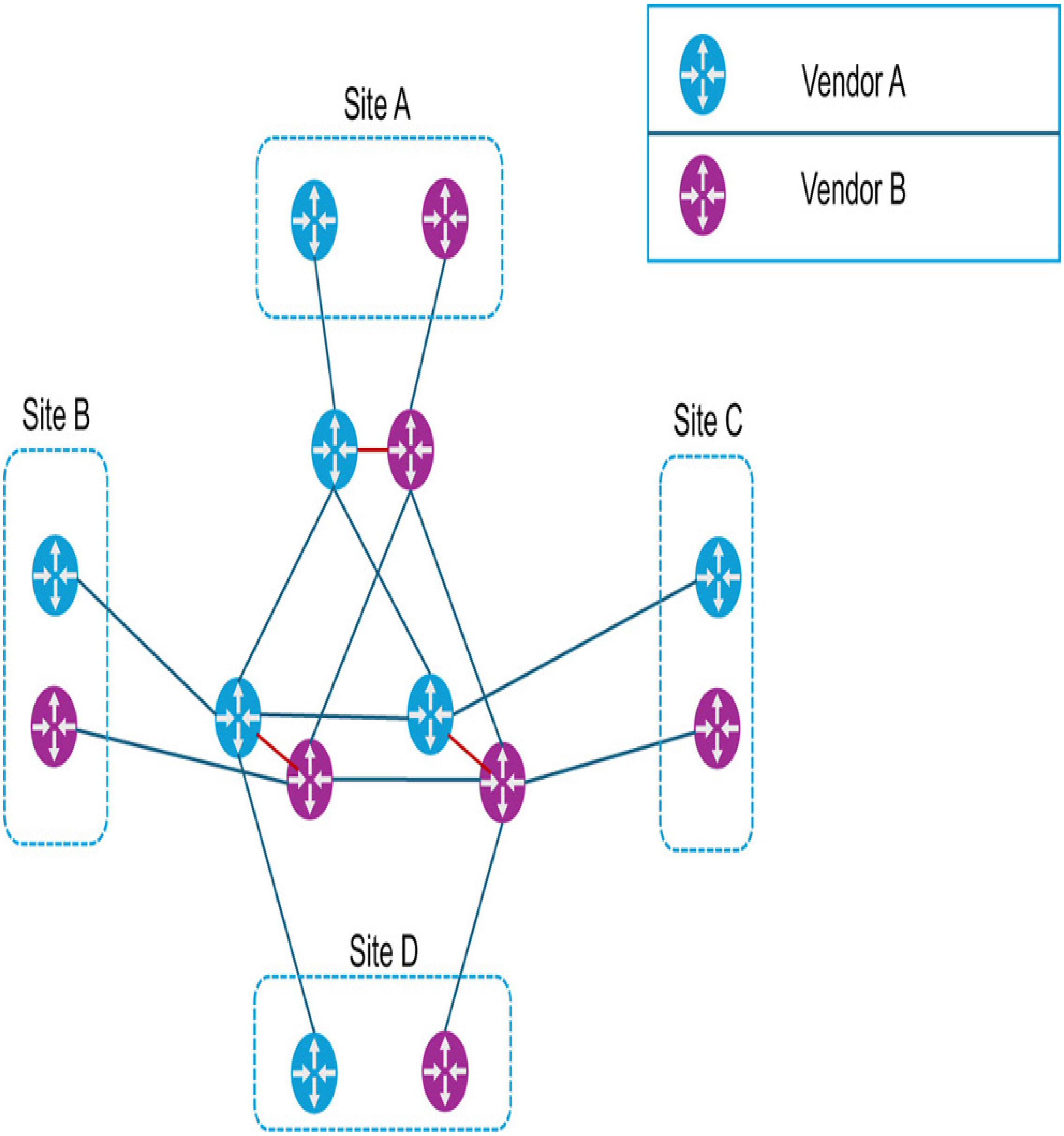


Figure 20-1 *Multivendor Wide Area Network Architecture*

This approach is not limited to just the physical network that represents the WAN; it is also considered applicable to other software- and network-related verticals.

Vendor redundancy can pertain to the resilience that can be achieved by deploying workloads in a standardized way across disparate data centers, cloud providers, or even IoT manufacturers to ensure that a single critical bug or problem will not halt critical lines of business.

Software Resilience

The foundation to a stable architecture is the software that runs on the systems themselves. Normally, there is a trade-off that needs to be considered between cutting-edge functionality and stability. While there is a chance that a brand new software version with many new features will be released and stable on Day 1, history has tended to prove otherwise; that heavy change revision in a production code base can lead to unexpected outcomes, particularly in the area of new development.

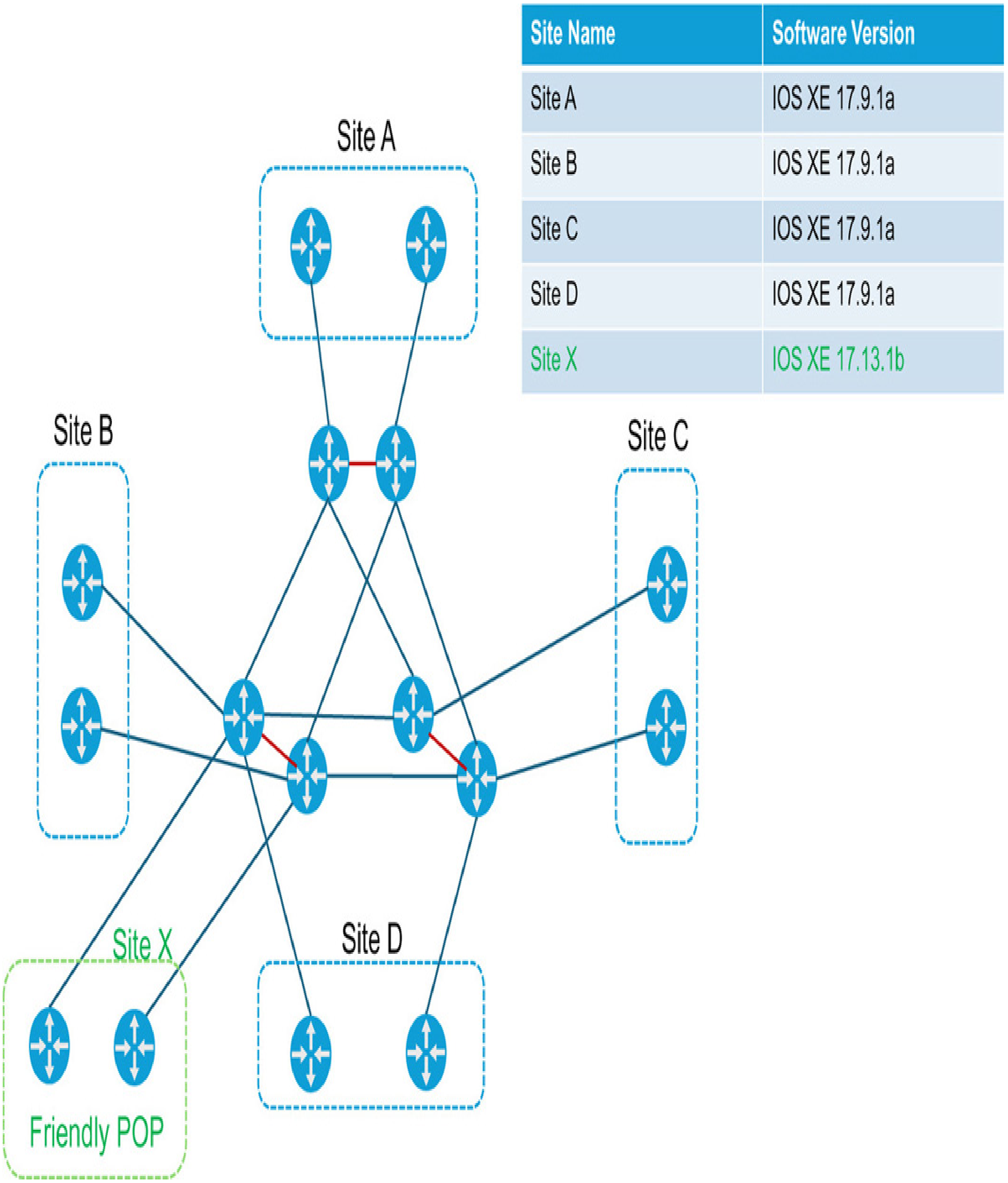
Software Versioning

Software strategy is a broad topic that is taken very seriously by the many customers who have customer-serving networks. How many upgrades take place per year, how much testing is performed prior to the upgrade, what the turnaround is for a security-relevant upgrade, and how long a new software version should be “burnt in” to the production environment are all relevant topics in this context.

Running the exact same version of software in an architecture could lead some to argue that the exposure to a specific bug or issue is higher. That being said, maintaining two separate releases may not always isolate the deployment against the same software defect or issues affecting both software release tracks/trains at the same time.

A strategy, particularly when deploying new software, and its rollout schedule require careful contemplation and planning. Normally, such planning requires a level of trade-offs be made; newer and less stable software versions tend to be software release tracks/trains that include cutting-edge features and functionality in contrast to long-lived software versions, which tend to have only incremental bug fixes and software limitations resolved, although largely lacking newer bells and whistles in terms of features and functionality, should they be required.

Figure 20-2 shows a strategy that is often pursued by service providers, where a friendly point of presence or friendly facility is used for validation of newer software images. The word *friendly* in this context usually means that the most critical (or loudest) users are not located on this site.



Site Name	Software Version
Site A	IOS XE 17.9.1a
Site B	IOS XE 17.9.1a
Site C	IOS XE 17.9.1a
Site D	IOS XE 17.9.1a
Site X	IOS XE 17.13.1b

Figure 20-2 *A Topology Deployment with a Selected Facility for Software Upgrades*

In-House vs. Commercial Off-the-Shelf (COTS) Software

When you’re considering further resilience within an organization’s estate, there are often trade-offs when it comes to business-relevant software that is selected. The issue is not always that the chosen software maps one-to-one with the business needs and requirements to map most efficiently to the processes associated with a respective line of business. While many vendors can provide off-the-shelf software that may match 80 percent of the needs, that final 20 percent often means that an extra level of customization is required. For some corporate software, the 80 percent is delivered as part of a core business application, with the final 20 percent being customizable and often offered as an extension or plug-in from a specific ecosystem partner.

Taking an alternative route, full in-house custom software development can meet 100 percent of an organization’s needs and, under certain circumstances, can offer software development cycles that are more rapid than those expected with commercial products that may exist in the open market. From our experience, however, this approach may show promise in the beginning, but technical debts can rapidly increase, resulting in a lock-in scenario for companies that decide to take this path.

A custom software lock-in scenario was observed with a social services organization in the Asia-Pacific that was providing benefits for its citizens. Forty years ago, this customer decided to take this approach in building out its own application ecosystem to exactly match with its needs to provide payment for social services, such as unemployment, retirement pay, disability pay, student support, and other benefits. While the custom software provided a high level of flexibility, over time, the code base and respective applications that the system offered became more complex and intertwined. With time, the initial programming languages that were chosen to develop this custom code, such as COBOL and C, became less popular, making it more difficult to find software developers to maintain the code base, and as the existing skilled workforce began retiring, this problem was only exacerbated.

As the industry shifted toward container-based microservice architectures, using systems like Kubernetes, the customer attempted to shift away from the monolithic software development structure that had been used for the previous decades through application refactoring. Although this approach had a lot of merit on paper, the attempt to refactor the applications proved very costly and was fraught with many challenges.

Eventually, based on the number of software developers needed to sustain the application ecosystem, the leadership of this organization concluded that the focus on the core business of social services was being hampered by becoming a pseudo-software company. The future costs associated with continuing this approach were no longer feasible. As a result, the organization decided on an alternative approach in shifting over to COTS applications.

This story is an extreme case but does represent an important reminder around technical debt and considerations that should take place when evaluating the correct approach for a given organization. If a significant amount of reworking and customization is needed for a core business application that maps to a domain or vertical that is very common, it makes sense to ask, Why is this the case? Is the organization doing something cutting edge or new? Are restrictive processes in place that have not been reevaluated over the years? Or does the industry simply work this way for the piece of software being used?

In addition to the challenges shared in this story, using COTS software does not necessarily mean that the solution is better. Contingency challenges in terms of software selection need to take into account the reputability of the vendor being used, and how much effort and time would be needed to swap out the application if the pricing is no longer appropriate or if the vendor were to go out of business. When you’re selecting software that interfaces with critical corporate data, considering such factors is important to ensure that the right business continuity and resilience factors are anticipated from the beginning.

Resilience in the Cloud

The deployment of resilient services is crucial for the survivability of key and critical services that are being hosted in modern data centers, both on-premises and in the cloud. The determination of what level of resilience is needed comes largely down to the service itself being offered. Does the service represent a critical component of a broader workflow that is dependent on it? Does the service have frontend visibility to consumers who may be working with a user experience (UX) that could result in a poor corporate perception if the service were offline? Is the service directly linked to revenue generation for the company, or it is just a “nice to have”? These sorts of questions need to be considered and answered when weighing the needs of having the service up and running, where it should run, and how resilient it should be.

When you’re looking at resilience in the context of reaching the cloud resources (public/private) over the network, several factors need to be considered. First, what communication path should be traversed to reach the service? Should the internal corporate network be used, including potentially fixed network circuits and links up to the last mile where the cloud service is reached? Or should the first egress point to the Internet be taken, traversing the Internet up to the entry point for the respective service?

In Figure 20-3, two options present themselves in terms of accessing the respective service in the public cloud.

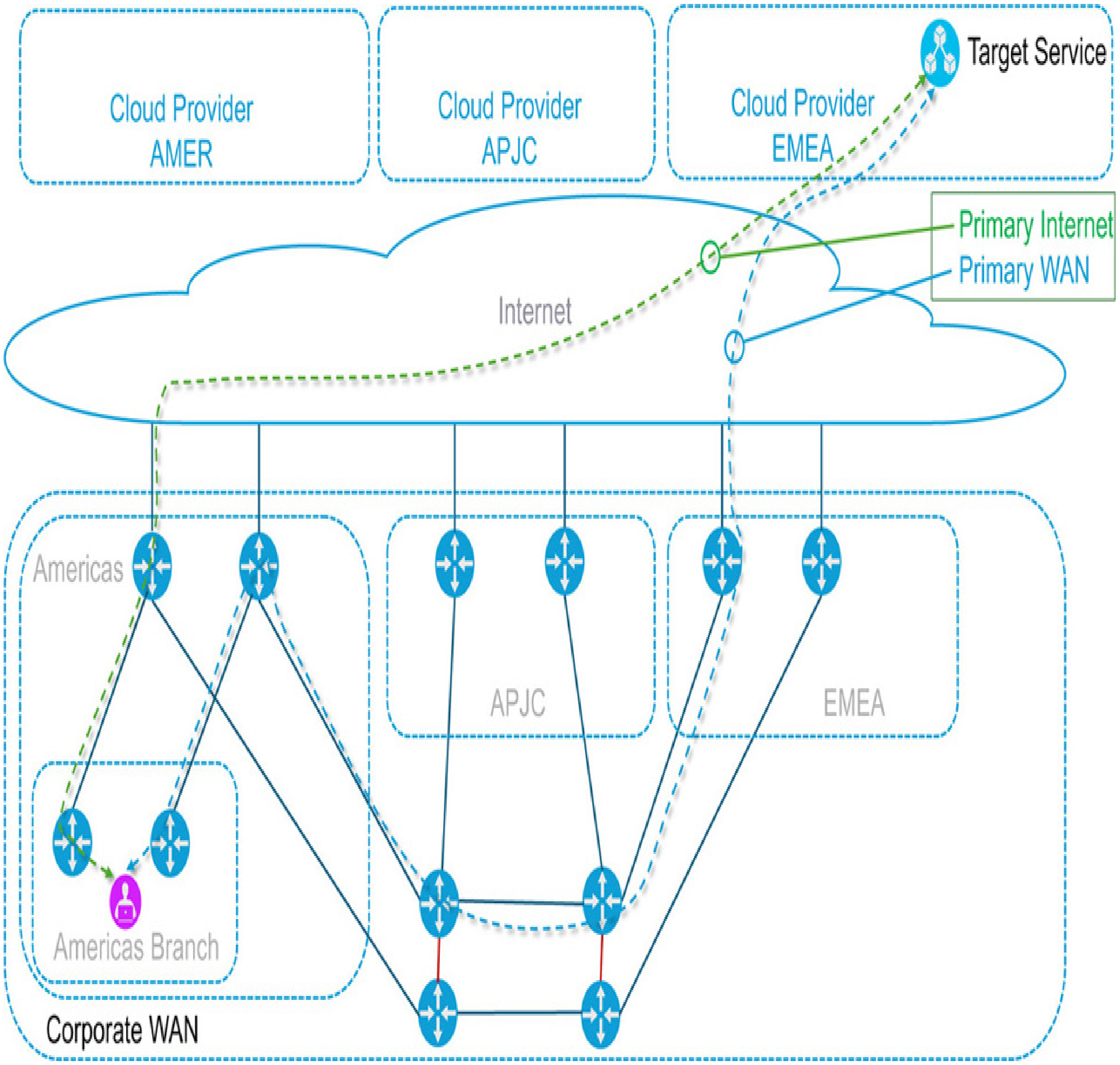


Figure 20-3 *Redundant Paths to Reach Cloud-Based Services*

In the figure, Primary Internet first utilizes the primary path through the public Internet originating in the branch in the Americas, taking the shortest path out of the corporate network to the Internet. Then the carrier connections that potentially traverse disparate carriers and BGP autonomous systems are traversed up to the point that the service is reached in the EMEA cloud provider.

Primary WAN then takes a path from the Americas branch through the corporate wide area network, finally egressing the corporate network to the Internet in EMEA, providing a shorter Internet path and proximity to the respective services to be reached.

In the context of resilience, the network that is depicted here does provide a resilient means to access the service that is being hosted in the EMEA cloud provider. The pros and cons with which approach to take in accessing the services tend to come down to reliability and observability of the communication flows. While the primary path to the service via the Internet may get the traffic out of the corporate network the most rapidly, avoiding congestion of links that may exist within the WAN, the limitation with this approach is often that the loss, latency, and other reliability of communications over the Internet natively are not as easy to track as a consumer, as is the path over a privately owned or managed wide area network. For this reason, many customers choose to hold onto communication flows within their corporate networks up to the shortest "last mile" to their respective cloud provider. Such an approach lowers the number of unknowns that may exist when having to traverse an unknown number of hops over the public Internet.

To further augment such connectivity, observability tools are often applied to key and critical services. Tools in this domain, such as ThousandEyes, Accedian, and Sam Knows, can provide a deeper level of insight into the quality of communications along a specific network path or even perform levels of synthetic testing to ensure that a clear view of service resilience is available. This allows for correlation between network issues reported and the paths traversed by client-to-server applications.

While observability tools can provide a visual context of where traffic is traversing and disruptions on the path, taking an approach to avoid a problematic path is the next step. Intelligent path selection has long been a capability within routing and switching software, starting with the use of source routing, which allowed clients themselves to decide on the path that they would traverse through an IP network. It quickly became apparent that this approach of placing the responsibility of the network traffic forwarding path in the hands of the user was not the best approach. Later iterations of event-based traffic steering came into play in large part through the deployment of MPLS traffic engineering, whereby strict (fixed hop-by-hop selection) or loose (intermediate-hop selection) paths could be configured for traffic going to a specific destination. This capability was further augmented with (Resource Reservation Protocol (RSVP), which provided signaling and bandwidth allocation for the respective path. While this capability greatly enriched the means by which service providers and customers performed routing in their network and, for secure traffic, were able to ensure strict traffic forwarding paths could avoid networks that may be considered insecure, it did not provide a means to ensure that the traversed links were without issues.

The introduction of SD-WAN with SLA-based path constraints provided the missing capabilities for many customer networks that were seeking assurances that their traffic would meet certain stability criteria: These criteria ranged from packet loss to latency requirements on the connected WAN links, circuits, or Internet breakouts leveraging tools such as Cisco Cloud OnRamp for connecting to central cloud Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

The main handoff options that customers tend to deploy are

- No SD-WAN: Direct Internet Connectivity Best Effort
- No SD-WAN: Direct Circuit termination to Equinix/Megaport
- No SD-WAN: Azure Express Route, AWS Direct Connect, Google Cloud Interconnect
- SD-WAN: Direct C8Kv termination in redundant availability zones
- SD-WAN: Direct C8Kv terminations in on-premises routers/virtual routers from Equinix/Megaport

Cloud service redundancy can be achieved by ensuring that key data and applications do not reside in a single location. This effort can be achieved in a number of ways—through the selection of deployment of key applications and services in different geographical regions to the deployment in disparate availability zones that are provided by the cloud provider.

In DevOps circles, site reliability engineering, and open-source communities, the term *platform engineering* has become popular over the years. It was initially coined by Gartner as a means to provide an abstraction layer between software developers and the complexities of the network, compute, and hosting environments. For many, the term simply refers to a simplistic and standardized entry point for software developers, abstracting the complexities of the infrastructure and hyperscalers, thus allowing software developers to spend their time on what they do best, which is to develop software rather than try to understand the nuances of each public or private cloud deployment.

Most platform engineering offerings are focused on leveraging an internal web-based developer platform, providing a self-service frontend, to spin up new development or production environments.

The deployment of a platform engineering architecture needs to be grounded in a number of key and core principles:

- Abstract the complexity of cloud (public/private/edge) provisioning and interaction from software developer teams
- Maintain visibility of the financial costs (FinOps) of the selected resources, providing clear accountability for timely decision-making
- Automatically include security tools when deploying new environments
- Automatically include application observability tools
- Follow software development best practices, deploying DEV, STAGE, and PROD environments

Over the years, a shift toward private and public cloud-based architectures has become essential for many businesses. The ease of entry, API-first-based mentality, and elastic resource allocation have enabled many businesses to ramp up, test, validate, and bring new services to market.

Some of the challenges that have arisen from the breakneck speeds that organizations have shifted workloads and resources over to the cloud, however, have manifested themselves in numerous ways:

- Lack of consistent security
- Lack of financial oversight
- Inflexibility in subscription models
- Increased costs
- Challenges in refactoring applications to meet needs in the cloud

- Difficulty in managing multiple cloud provider estates

The problems described here tend to be very familiar when talking with customers. In fact, Cisco IT also experienced many of these challenges within its own estate.

March 2020 was the beginning of the COVID-19 pandemic, the highly contagious virus resulting in global waves of fatalities, and government mandates for people to lock down, remain within their homes, avoid gatherings, and only leave their premises for essential reasons. The consequence that these new rules and restrictions had for global businesses was profound. Many organizations had never prepared for a remote working strategy, lacked the security architectures for remote access to their network, and also lacked collaboration tools that would support remote working.

As a vendor for collaboration tools such as Webex and security products that support remote access, Cisco observed an unparalleled increase in customers wanting to leverage these services. As a consequence, Cisco IT teams scrambled to deploy compute architectures that would facilitate the rapid ramp-up of capacity for these services. Figure 20-4 is a snapshot in time of the Cisco IT global architecture, including its corresponding data center and cloud.

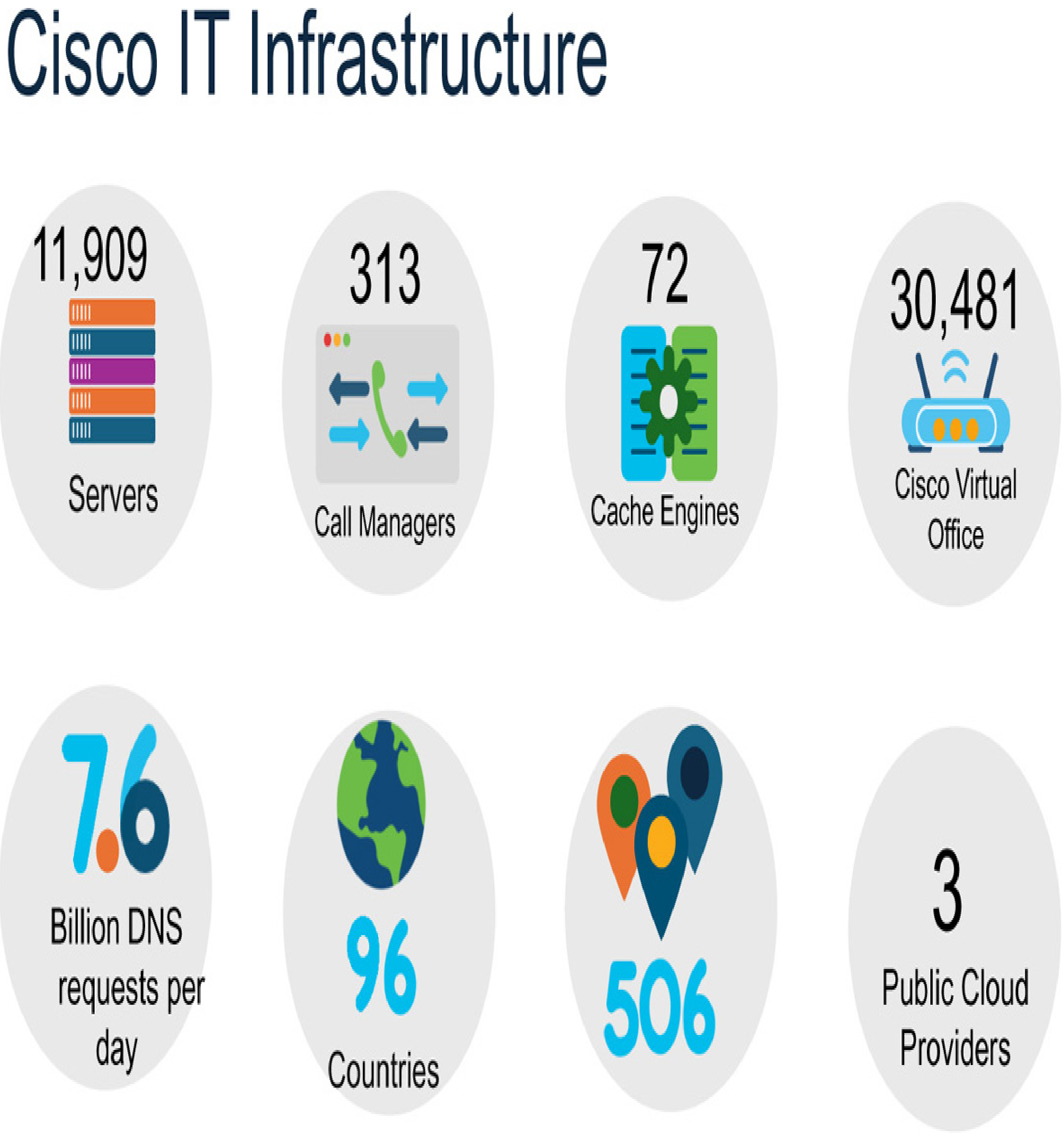


Figure 20-4 *Snapshot of Cisco IT Global Services Across the Estate*

While Cisco did manage to support customers in their goals to ramp up new security and collaboration services, the haphazard panic that resulted in the services being generated led to scenarios that would take time to remediate. The rapid deployment in cloud and on-premises data centers, while achieving the goal of bringing customers online, lacked the planning, foresight, and operational rigor that would be applied to a properly planned increase in service. As a result, services were stretched across multiple regions and multiple hyperscalers (and on-premises), and in some circumstances, corporate credit cards were even used to purchase cloud services.

In the years that followed, the entropy introduced via the chaos of the pandemic had to be reined in. The best way to achieve this was to shift existing services that had sprung up rapidly to pivot toward a platform-engineering approach. The advantages awarded by this approach were the ability of developers to maintain access to the key tools and services that they had grown accustomed to using. However, the right standards, including baseline resilience and security, were deployed by default, leaving the job of understanding the nuances of cloud-based security rulesets to the platform engineering teams, and allowing software developers to focus on doing what they do best—develop software.

Consequences of Authentication and Authorization Resilience

Ensuring a secure estate involves taking the right precautions to allow for outage scenarios, future growth, and potential scale scenarios that can happen during upgrade and provisioning scenarios. Considering a worst-case scenario often helps in identifying the weakest links in the chain, and preemptively making the decisions around the trade-offs that may exist between cost to invest in infrastructure versus cost and impact to the business in the case of a loss of service.

Deploying a robust infrastructure that validates the identity of the user to provide selective access to resources is one of the key cornerstones of a zero trust–based network architecture, which is introduced in detail in Chapters 1 through 4 of this book. The premise of such a deployment is that the architecture that can provide such validation and determination of role, rights, and access is available. Best practices for deployment of such an infrastructure can vary, with considerations that must be made for how resilient the architecture should be, based on the criticality of the infrastructure, services, or applications that are being offered.

For instance, systems that provide network access to communicate with 911 emergency response operators clearly have a critical need to be up and running all the time. If a situation impacts the infrastructure, there should be clear procedures in place to support calls from an alternative regional operations center. Online webstores may also have a high need for authentication redundancy because there is a direct and tangible impact to the number of sales that can be made if users are not able to log in to conclude their purchases.

In the context of network infrastructure—in particular, TACACS+ and RADIUS—a priority list of methods can be selected to decide which order of operation is taken for a given type of authentication and group of servers. Such deployments can be augmented through the deployment of load-balancing infrastructure to ensure that systems can be taken offline for maintenance activities and more intelligent monitoring that could result in a server being taken out of use when it is considered unhealthy.

For AAA infrastructure, in large deployments, Cisco's recommendation is almost exclusively focused on the use of load balancers. This recommendation is based on years of experience in customer environments, where simplistic comparisons can be made around the consequences of attempting to balance the load across such infrastructure manually.

Many operators may believe that it is possible to manually assess load based on datasheet numbers and evaluation of potential outage scenarios. However, history tends to show that such evaluations are point-in-time assessments, which do not consider the constant growth and change in user and usage patterns that take place over time.

Figure 20-5 depicts various options for resilient deployment of RADIUS in an authentication infrastructure. The components of the ISE RADIUS infrastructure shown in the figure consist of policy administration nodes (PANs), which are used for policy administration, device management, and main system settings; along with monitoring and troubleshooting nodes (MNTs), which are responsible for alarming and event notifications, logging and compliance, and monitoring and reporting functions. Finally, policy service nodes (PSNs) are responsible for policy enforcement.

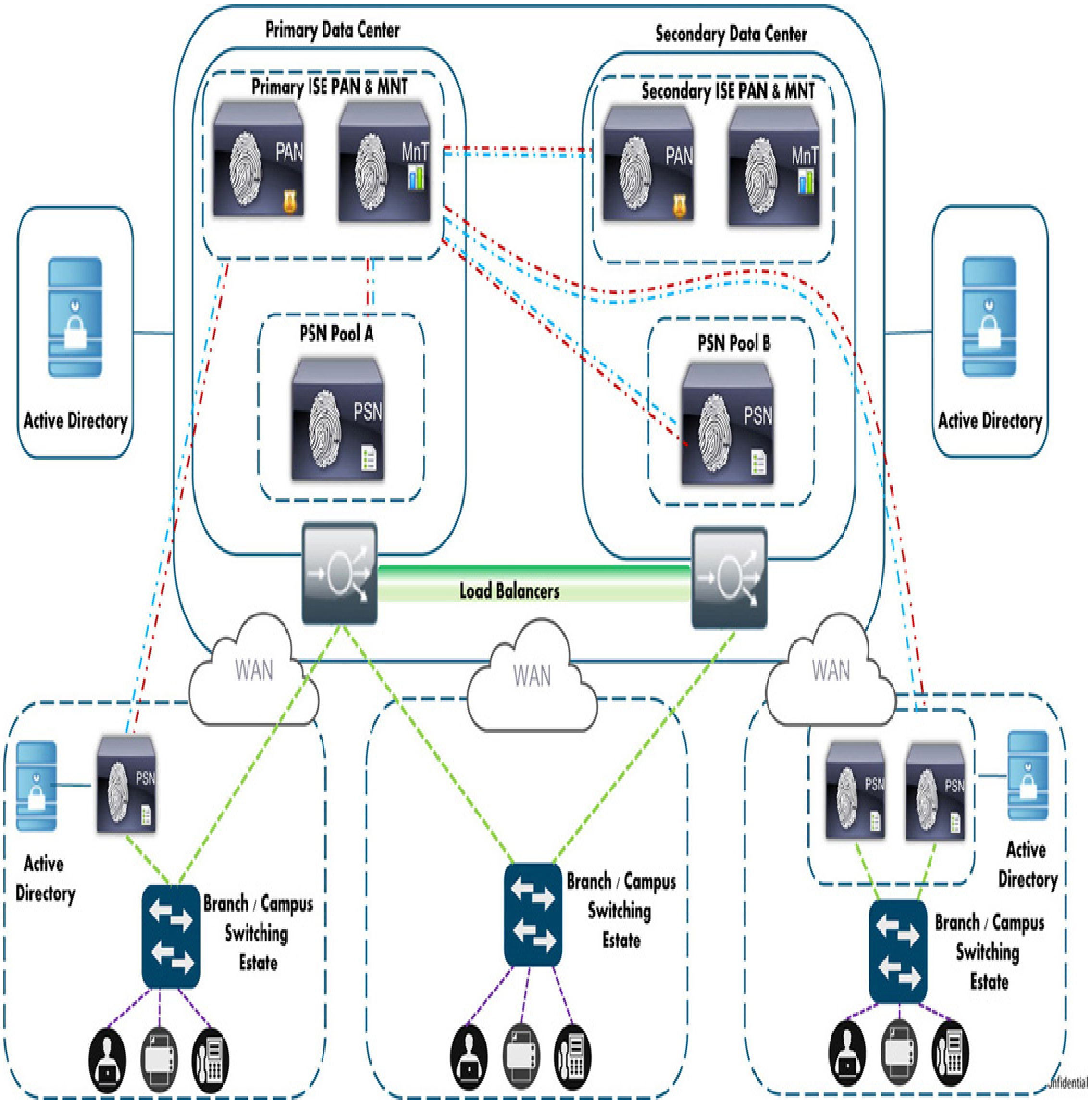


Figure 20-5 *Redundant Radius Architecture Deployment (Cisco ISE)*

For the middle construct shown in the figure, where there is no direct on-site connection to a local RADIUS server and its connected identity server, the security team will need to make some tough calls when it comes to how the switching infrastructure on the site should respond in the case of a failure of connectivity:

- Do existing authenticated systems remain online?
- Should authenticated systems be taken offline after a particular period of time?

- How should new connections be handled? Should access to printers be used?

Usually, such assessments take place within the planning phase and are performed in conjunction with a risk assessment, identifying what the real impact would be in scenarios where the authentication infrastructure may be unreachable or offline. The answers to these questions do not fall into a one-size-fits-all category. Answering them is rather a careful balance of pros and cons relevant to the associated organization and its core business functions.

For network devices that leverage the TACACS+ protocol for authentication, authorization, and command accounting, dedicated server architectures often are selected during the design phase of a project. This is usually performed for the following reasons:

- Decoupling of RADIUS and TACACS+ personas to avoid combined impact during upgrades
- Decoupling of identity store mappings between TACACS+ and RADIUS to avoid login credentials and system or machine authentication credentials being common

Much like the deployment of a redundant RADIUS architecture, a redundant TACACS+ architecture can help ensure that valid users are accessing the systems and that their interactions are appropriately protocoled. Depending on the configured setup, the further option of directed requests, which allows users to select their TACACS+ server during authentication, can support specific use cases that may be necessary to achieve a larger-scale deployment and further methods for authentication use if the primary servers are not reachable.

In addition to the server-based infrastructure reachability, there is also the possibility to configure local authorization rules that are linked to a privilege level of users who are locally configured on the system for authentication. This capability can be useful for scenarios in which the TACACS+ infrastructure may not be reachable. The limitation with this is largely associated with the maintenance of a viable centralized audit trail.

Many of the components that we’ve mentioned so far are related to infrastructure authentication. In most scenarios, there is a link between the infrastructure user repository and backend systems such as Active Directory, Entra ID, or other sorts of identity stores. Simply providing a redundant set of AAA infrastructure components without ensuring that the backend user repositories and databases (such as the CMDB) are also resilient will fail to achieve the needed objectives and goals. Resilience in an IT infrastructure is only as robust as its weakest link.

Taking measures to ensure that a well-thought-out structure is in place to support the end-to-end identity journey and its components is key to ensure that the architecture can weather outages, upgrades, security notices, and other symptoms that may be well beyond the operator’s control.

In the context of identity, systems that often need consideration for their resilient deployment are

- Microsoft Active Directory
- Microsoft Entra-ID reachability
- Certificate and PKI infrastructure
- MAC address databases
- CMDB inventory asset databases
- NTP reachability (to ensure certificates are valid)
- Multifactor authentication systems (DUO, etc.)
- Identity intelligence systems
- Asymmetric encryption keys and dependent servers
- Federation identity providers (SAML/OpenRoaming)

Client and Server Agent Resilience

As mentioned previously in this chapter, resilience is only as strong as the weakest link in the chain. In an IT ecosystem, that can be a very long chain indeed. Considering all the touch points that today’s businesses need to engage with to ensure a robust and secure estate, supporting core business applications and functions is not a simple task.

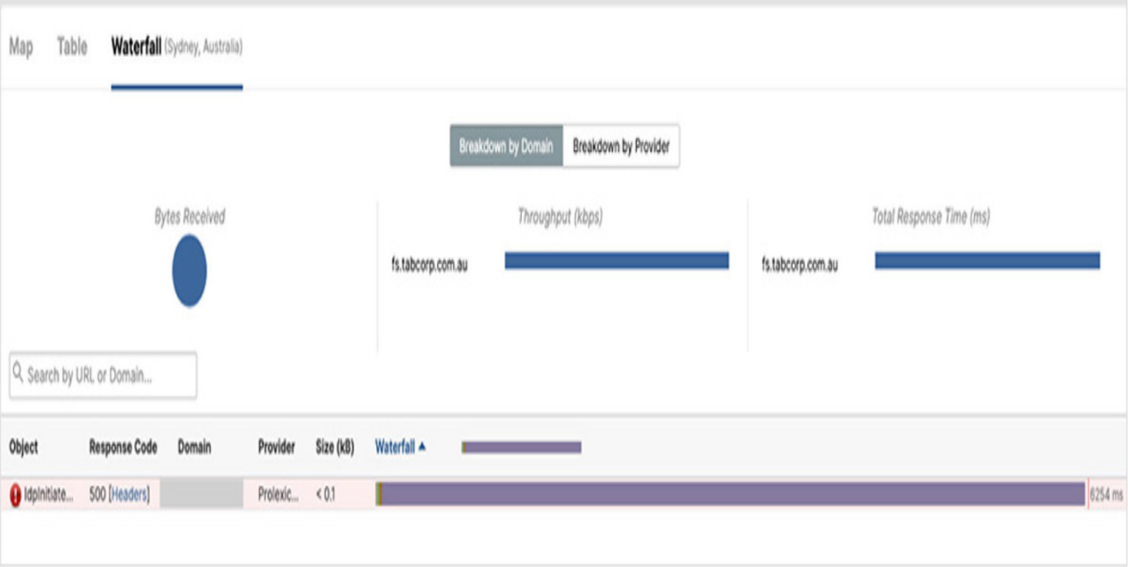
One domain that is often misunderstood, or forgotten, is the process of the agent software that is often running on the machines themselves. Client-side and server-side agents have become a common part of network ecosystems, ranging from security agents on client endpoints, such as Umbrella, Cisco Secure Client, ThousandEyes, or even client- and server-side tools such as Zabbix, which maintains a diagnostic view of the state of an endpoint.

Most of the agents mentioned previously are network specific. When we add on top all the agents that workstation build teams focus on to ensure proper remote support and management, such as mobile device management (MDM) tools, the ecosystem that is in use starts to become very diverse and very broad.

Many network operators consider the testing and validation exercises that are performed in the infrastructure as a relatively narrow activity that should be limited to only the network infrastructure components to properly validate the robustness of the solution. Similar levels of testing and validation are also critical on client endpoints and servers. Unlike the upgrade of a Cisco device, which utilizes a common and fixed manifest of hardware to function with its software, client and server teams face a constant fight for functional compatibility with different builds of Linux, Windows, macOS, and other server-side software. It is not enough for a system to be able to boot up and load a word processor on these devices anymore. Modern systems need to function with a careful mix of drivers for Wi-Fi, agents for security, and hypervisors for specific applications used, and they must maintain functionality and consistency with these components after each and every upgrade.

To achieve a functional and stable architecture within the client and server domain, you need to apply similar principles as to what are common in Infrastructure as Code and software development domains. For each new release, automated testing cycles need to be executed, with only valid, tested, and working releases making their way into production. A failure to perform these actions can become quite costly and result in a significant impact to an organization’s lines of business when they are not achieved.

One example illustrating a failure to properly perform prevalidation of software prior to rollout resulted in a critical state on key and critical applications; this occurred with the Microsoft Windows Server Security Agent from the CrowdStrike managed detection and response (MDR) service. The faulty update that was pushed “triggered a logic error resulting in a system crash and blue screen (BSOD) on impacted systems.” This error affected systems globally, including that of the Federal Aviation Administration. Flights were grounded across the United States due to the impact observed by various airlines. The wide-reaching impact affected healthcare, commercial, logistics, and financial services. The ensuing impact of the outage could be observed through server timeouts and server errors represented by HTTP 500 messages in scenarios where frontend applications could not reach their backend services, as shown in Figure 20-6.



Object	Response Code	Domain	Provider	Size (kB)	Waterfall ▲
Idpinitiate...	500 [Headers]		Prolexic...	< 0.1	6254 ms

Figure 20-6 *View of Affected Online Service During Outage*

Other critical applications, shown in Figure 20-7, also simply did not respond as a consequence of traffic dropping at the entry point toward the service.

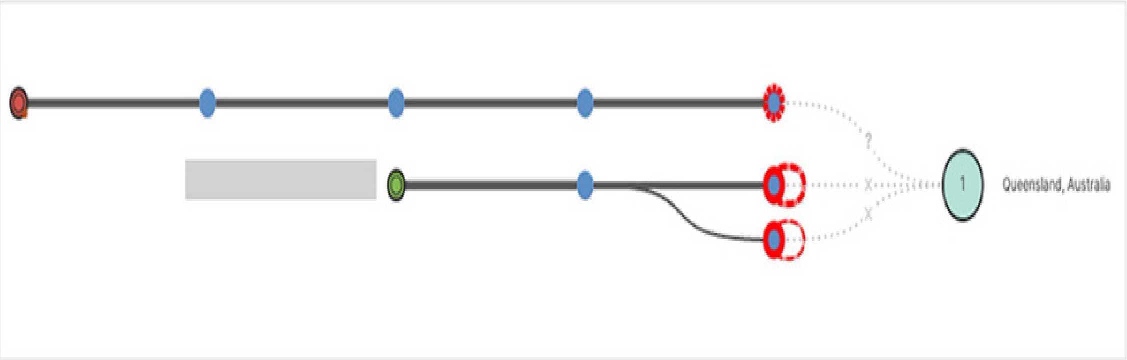


Figure 20-7 *ThousandEyes Dashboard Capture of Affected Paths to the Service*

The impact of the outage shows the consequence of not performing the right levels of testing and validation before rolling out updates on critical systems. Confirming the update checksum (to ensure that the update is indeed originating for the expected source), deployability, and post-deployment health on reference testing architectures prior to deployment in production is a key aspect of ensuring a functional, stable, and working environment. When services are so critical that downtime cannot be tolerated, consideration should also be made toward variation in software versioning selection or even product selection to ensure that an impact like the one described here will not have far-reaching effects across the entire estate.

Audit Trail Resilience

An audit trail enables network operators to provide a historical record of the actions that took place at a given time in the past. It serves as a valid means to troubleshoot and perform diagnostic follow-up after a disruption or an outage, but it can also support activities associated with identifying which actions led up to a particular security breach or who might have been active in a particular network or utilizing a particular resource in the past.

Depending on the country and the usage, retaining audit trails—much like tax returns—may even be legally required. In numerous countries that offer free public Internet access, it is not uncommon that a combination of username, MAC address, MSISDN (telephone number), and Network Address Translation (NAT) session information—sometimes delivered via Network Secure Event Logging (NSEL)—needs to be maintained as a persistent record if the authorities need to track down an individual who was accessing or transmitting illegal content, for example.

What exactly is an audit trail? An audit trail could be several different log files or outputs that can lead to the conclusion of events that happened. Examples are

- Security appliance rule hit logs
- Authentication logs (login/logout)
- File access logs
- Network Address Translation logs
- Network traffic logs
- Configuration change logs
- Application access logs
- Cisco TrustSec deny logs (applied in policy)
- Core dumps from infrastructure devices

These different logs all represent historical events that may have a short-term purpose in validating the overall health of a system or mid- to long-term usage in the context of forensic analysis or as a means for historical benchmarking.

How can the logs that exist in an audit trail be trusted? Placing all relevant and critical logs onto a single system with common access rights is clearly not considered the best approach to ensure that the data is secured and not manipulated. When you’re deploying critical logs, a redundant approach should be pursued, providing three streams of log data.

Preferred Location Resilience:

- Location A: Read Only: Accessible to operators who are performing triage
- Location B: Read Only: Accessible to systems that may use the data for visualization (Splunk, etc.)

- Location C: Read Only: Artic Storage, backed up and maintained for future recovery or use where necessary

A further best practice for the described audit trails and logs is that the systems are all configured with a minimum of three NTP servers to ensure that they maintain a common and standardized time source. This is the minimum number of servers needed to help identify if the clocking skew is off on one of the three devices. If only two servers are configured, identifying which device is incorrect is not trivial.

Audit Trail Reputability

As described previously, logging information that represents an audit trail should be deployed in multiple locations to ensure that there are backups that can be accessed in the case of a catastrophe. In addition to the ability to make backups of the data, it is important to have a means to avoid tampering and provide a mechanism to identify tampering of the logs in the trail if it does happen. Where a resource or component within a data center has been breached, if only one system has the data stored, it lacks the capability to compare with other versions of the data.

While the introduction of redundant locations for audit trails and logging data can be helpful in increasing the likelihood of integrity, if a breach occurs, and common levels of access to the audit trails are available across the estate, it is quite possible that the other copies of the logs have also been edited. This is why different data storage methods, with different levels of user access rights, become relevant.

Under certain circumstances, high-security environments can also consider storing such data using blockchain-based file system methods, such as InterPlanetary File System (IPFS). Despite the interesting name, the deployment of such an architecture, whereby the data is maintained across distributed systems and its existence and data are maintained on a shared ledger, can help provide nonrepudiation for the data that has been deployed.

Figure 20-8 shows how data can be stored with integrity, via an IPFS type deployment, resulting in a cryptographic hash being applied to uploaded files, whereby the data is replicated across participating nodes.

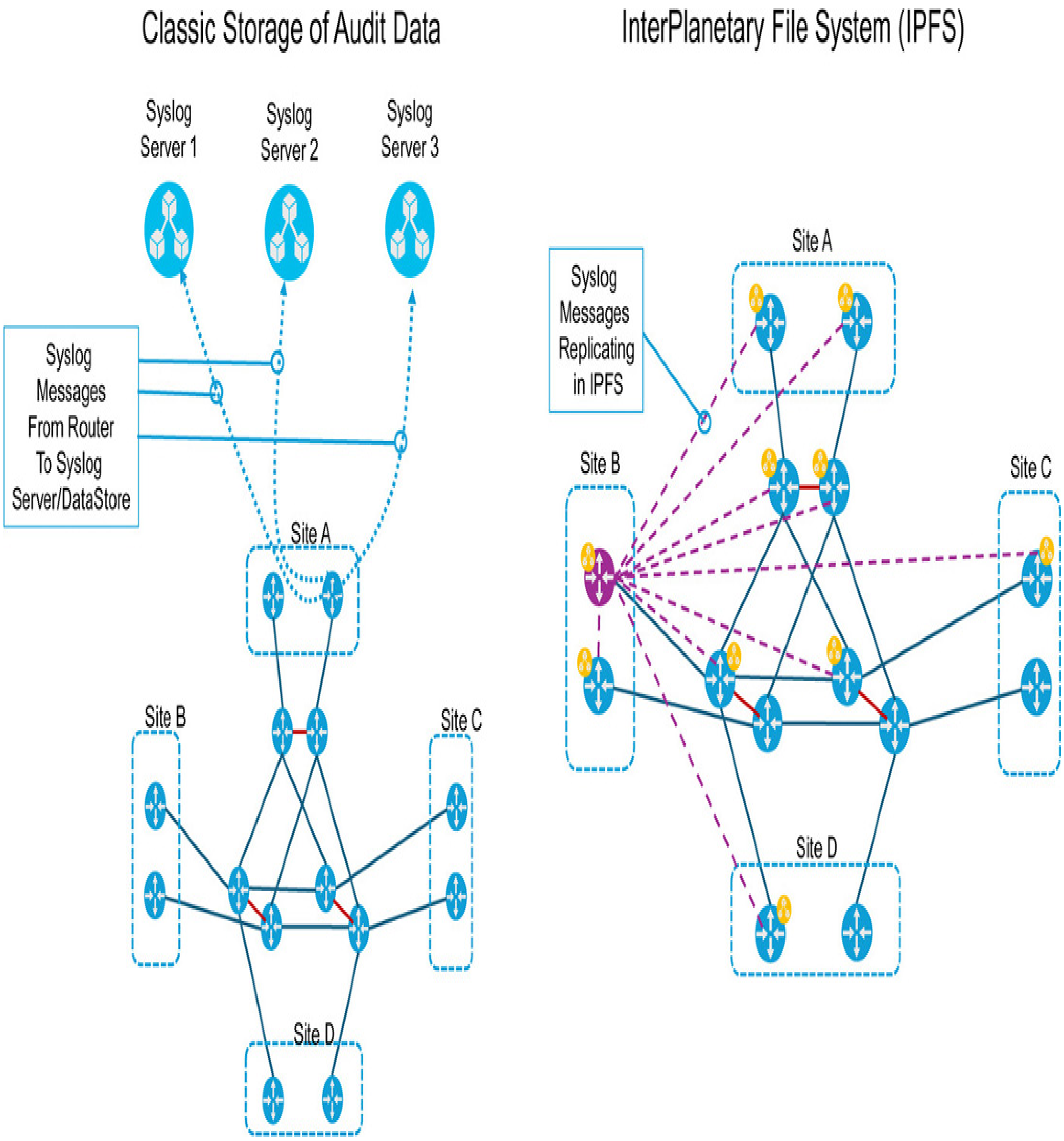

Classic Storage of Audit Data
Syslog Server 1
Syslog Server 2
Syslog Server 3
Syslog Messages From Router To Syslog Server/DataStore
Site A
Site B
Site C
Site D
InterPlanetary File System (IPFS)
Syslog Messages Replicating in IPFS
Site A
Site B
Site C
Site D

Figure 20-8 *Using IPFS to Achieve Distributed Storage Using Web3*

While this approach is technically feasible and Cisco has applied it for customers as part of professional services engagements, it is not part of the default IOS XR, IOS-XE, or NX-OS file system behavior.

Although the use of blockchain-based storage is not considered the norm today and is often not even considered given the close connection that individuals make between bitcoin and blockchain technology, its capabilities have started to become more popular over the years, with its use becoming more prevalent in merchant banking sectors. In some even more advanced scenarios, blockchain technology is used to ensure robust trackability of corporate supply chains, as is the case with the Tracr application used by De Beers to allow cradle-to-grave tracking of diamonds; or in the case of Estonia, where a significant amount of public services shifted over to the use of blockchain as a means to ensure reputability and a robust and trackable audit trail through a common distributed ledger.

Reliability of Auditable Data

The presumption of a valid audit trail comes down to factors beyond the transmission of the log or trace data toward its target destination. There is also the need to ensure that the generated data can actually leave the platform in the first place. There are numerous attack paths, whereby the creation of a core dump or log data is blocked from ever being created or leaving the system. This scenario is often the result of more complex exploits being exercised, whereby in-memory implant techniques are used. Such scenarios have been observed in the past few years with the ArcaneDoor and Line Dancer exploits, where a hacker disabled syslog on the exploited system and performed a hook in the crash dump process to prevent core dumps from ever being written.

Thankfully, such advanced exploits are not particularly common today, but they do provide a glimpse into the challenges that can exist around audit trail data integrity, and potential justification in very secure domains for the use of multiple platforms or multiple vendors in parallel to raise the complexity of network or system entry.

Proactive Resilience Validation

Historically, in many environments, redundancy and resilience testing for infrastructure and software was performed during initial integration. With Cisco Network Infrastructure, the testing phase was often referred to as Network Ready for Use (NRFU) testing. These tests tended to contain the different links, interfaces, processes, nodes, and protocols that required disruption to confirm that a redundant secondary or tertiary path was indeed valid and functional. The challenge that many vendors noticed was that, once performed, that testing would never take place again. Many organizations lived off the false assumption that when a solution was validated and tested during the initial integration years, nothing changed, and the validated resilience was still in place and working.

Many of Cisco's peers in the software industry also observed these challenges during software development activities, which spurred the creation of *chaos engineering*. Beginning early on with Apple building functions and logic within its testing procedures to stress-test its MacWrite and MacPaint applications, it utilized a small software application called Monkey that performed randomized and rapid input to the applications in the hope of generating some form of instability in the code that could later be isolated and fixed. Over the years, the domain of chaos engineering grew beyond the application world for use by hyperscalers such as Google and Amazon, whereby the randomized restart of services and hardware systems would take place to attempt to spot gaps in resilience. Figure 20-9 shows what the chaos engineering process looks like.

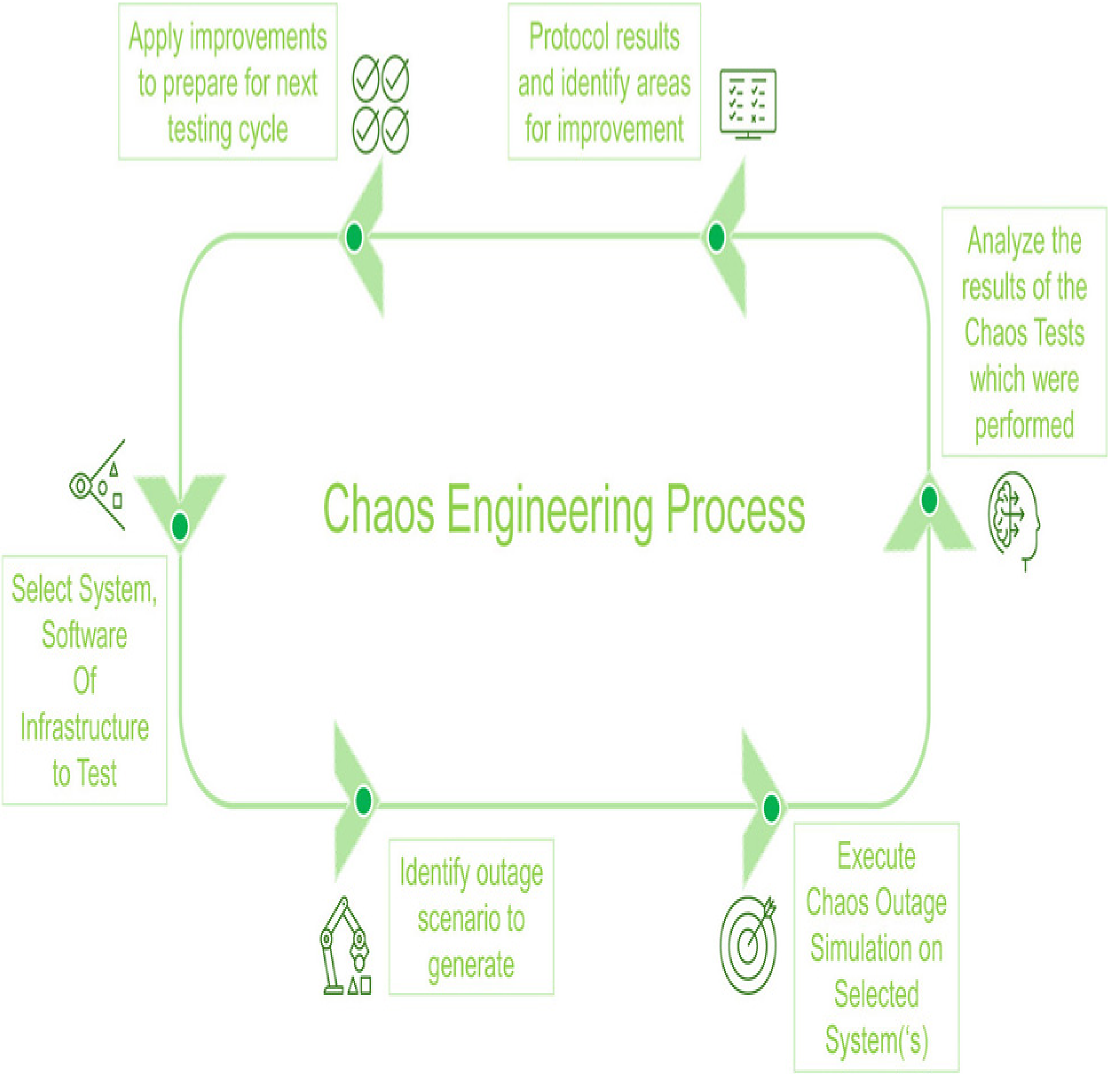

Apply improvements to prepare for next testing cycle
Protocol results and identify areas for improvement
Analyze the results of the Chaos Tests which were performed
Chaos Engineering Process
Select System, Software Of Infrastructure to Test
Identify outage scenario to generate
Execute Chaos Outage Simulation on Selected System('s)

Figure 20-9 *The Chaos Engineering Process*

Numerous organizations have begun to pursue this approach in the area of IP networks; however, it is not without its challenges. Rebooting an application process that allows remote access to restart it is not a serious risk. In contrast, rebooting a customer-side router or switch in a remote site that may lack out-of-band management and out-of-band power control is often perceived as being significantly more risky.

The approach itself makes a lot of sense and does lead to a more robust architecture, but if you want to perform such activities, the right prerequisites need to be available to allow for recovery in scenarios that could occur when a chaos engineering–based resilience activity is pursued and executed.

Network Infrastructure Resilience Consideration

Over the years, Cisco and its professional services teams have been involved in many customer architectural conversations and design sessions. One key lesson learned in these sessions—and from experience—is to always be prepared. Unless the architecture is a lab—and even in such a scenario—sometimes having the right levels of redundancy and resilience planned can be critical.

Following are key considerations that should be remembered when planning:

- Every device should be deployed with dual power supplies.
- Dual power supplies should connect to separate power entry grids.
- Every WAN router should be connected to two ISPs.
- Fault tolerance should be properly planned (no single points of failure).
- First hop redundancy protocols should be considered when needed.
- Redundant links and routed paths should exist (e.g., MPLS, BGP, ECMP).
- Recovery time objective (RTO) and recovery point objective (RPO) should be identified.
- Disaster recovery planning must be aligned with the business continuity plan of the organization.

If the right checks and balances are maintained during the planning and operation phases, it is possible to run and maintain a reliable IT architecture.

Summary

The deployment of a resilient architecture has many components, ranging from the physical location of components within a building to redundant WAN or Internet handoffs or connection mediums, to the selection of redundant cloud or on-premises data center ecosystems. As the reliance on technology becomes more and more ubiquitous with daily business and personal activities, it is important to focus not only on achieving function but also on achieving resilient functioning within the application, infrastructure, and security audit and infrastructure audit trails that are relevant for today’s IT ecosystems and environments.

Chapter 21. Zero Trust in Industrial Manufacturing Vertical

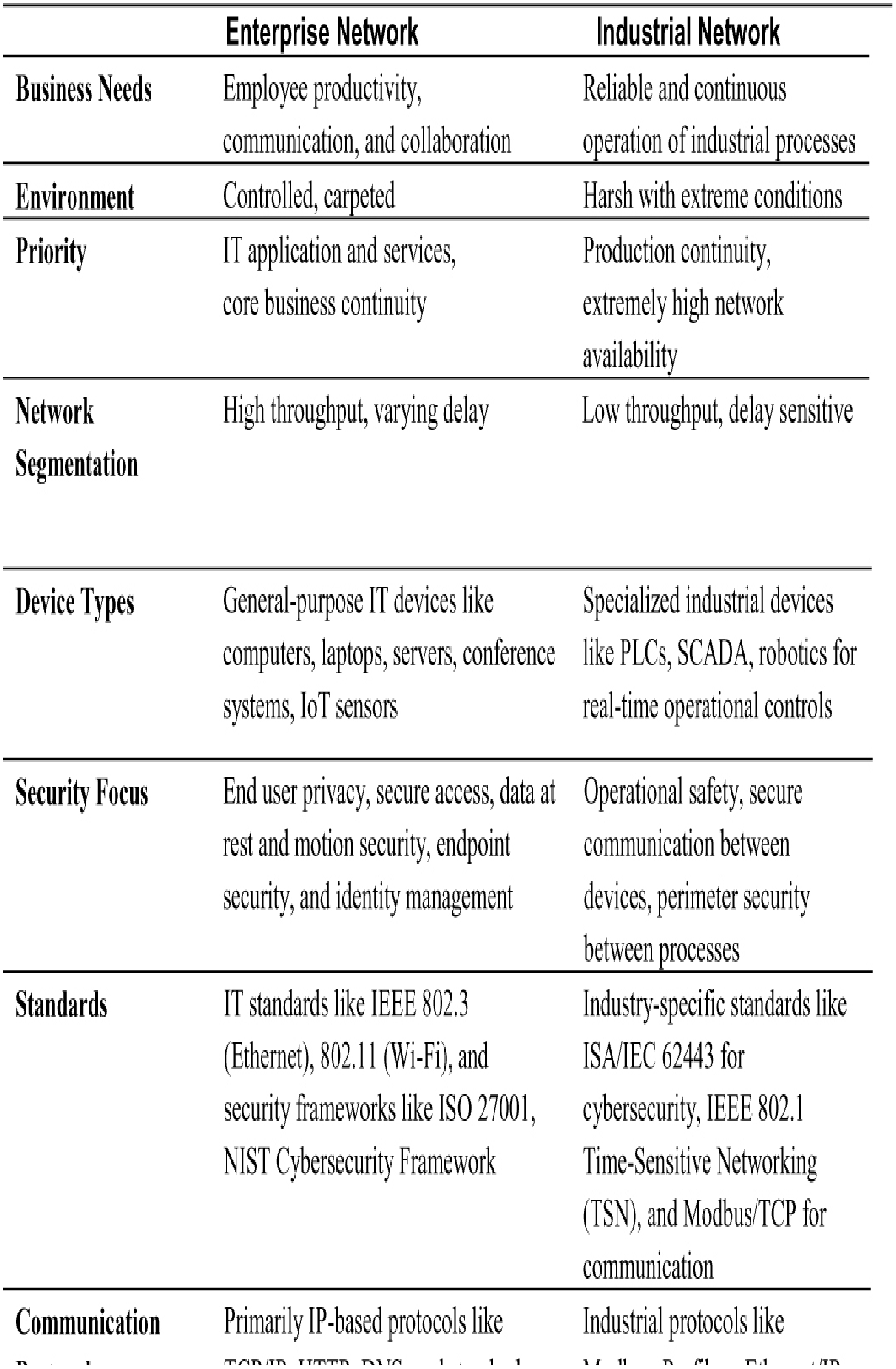
In this chapter, you will learn about the following:

- What makes zero trust implementation different in a manufacturing vertical
- The Purdue Model and segmentation constructs in a manufacturing plant
- Visibility and policy-based control for plant communication
- Secure remote access with ZTNA
- How to extend ZTNA in an extended enterprise environment with Cisco SD-Access

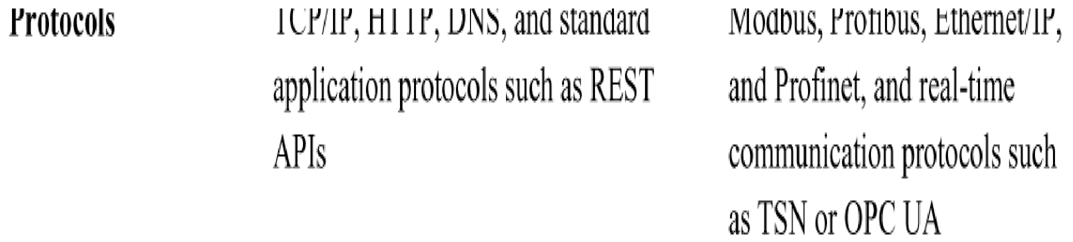
Introduction to Industrial Networking

Manufacturing plants fall under the category of industrial networking and security. Industrial networking is different from enterprise networking in many ways. You can think of them as a parallel universe to enterprise network setups. While the industrial network is meant to provide connectivity to industrial machines and other components, the environment, business needs, and priorities are very different. Table 21-1 compares the enterprise network setup with a typical industrial network setup.

Table 21.1 *Comparison of Enterprise and Industrial Networks*



	Enterprise Network	Industrial Network
Business Needs	Employee productivity, communication, and collaboration	Reliable and continuous operation of industrial processes
Environment	Controlled, carpeted	Harsh with extreme conditions
Priority	IT application and services, core business continuity	Production continuity, extremely high network availability
Network Segmentation	High throughput, varying delay	Low throughput, delay sensitive
Device Types	General-purpose IT devices like computers, laptops, servers, conference systems, IoT sensors	Specialized industrial devices like PLCs, SCADA, robotics for real-time operational controls
Security Focus	End user privacy, secure access, data at rest and motion security, endpoint security, and identity management	Operational safety, secure communication between devices, perimeter security between processes
Standards	IT standards like IEEE 802.3 (Ethernet), 802.11 (Wi-Fi), and security frameworks like ISO 27001, NIST Cybersecurity Framework	Industry-specific standards like ISA/IEC 62443 for cybersecurity, IEEE 802.1 Time-Sensitive Networking (TSN), and Modbus/TCP for communication
Communication	Primarily IP-based protocols like	Industrial protocols like



Protocols	TCP/IP, HTTP, DNS, and standard application protocols such as REST APIs	Modbus, Profibus, Ethernet/IP, and Profinet, and real-time communication protocols such as TSN or OPC UA

It is important to understand the difference between the terms *information technology (IT)* and *operational technology (OT)*. IT refers to the digital infrastructure of an organization that ensures secure data transfer over the designed computer network. It is designed to provide secure connectivity over wired or wireless mediums. IT enables communication between machines, servers, and the Internet. OT, on the other hand, focuses on providing hardware, software, and network connectivity to the industrial environments. In simple terms, OT is the area where actual production happens within a manufacturing plant. Typical endpoints you will see in an OT area include programmable logic controllers (PLCs), supervisory control and data acquisition (SCADA), and other networked devices like robotic machinery that automate and control machinery, ensuring real-time communication and reliability for critical operations.

The OT area of a manufacturing plant is a harsh environment characterized by high temperature, humidity, dust, vibration, and so on. In such environments, you cannot deploy enterprise-grade networking devices like routers, switches, Wi-Fi access points, and firewalls. You will need ruggedized industrial networking devices that can withstand harsh environments. We IoT architects humorously label enterprise networking gears as executives with privileges, because they enjoy the pristine environment of a data center with air conditioning, whereas IoT/OT networking devices are like saints doing meditation in harsh environments with extreme cold or hot. Cisco has a wide range of industrial networking devices, many of which are purpose-built for specific industrial environments. They are IP67-rated, with no moving parts such as fans; instead, they have an internal heatsink. These devices support specific industrial protocols like PROFINET and Common Industrial Portfolio (CIP). For redundancy and loop avoidance, you will use protocols like Resilient Ethernet Protocol (REP) and Media Redundancy Protocol (MRP) instead of Spanning Tree.

In this chapter, we will focus primarily on industrial switches and firewalls because they form the building blocks of zero trust architecture. Figure 21-1 shows some of the devices from the Cisco Industrial Portfolio.

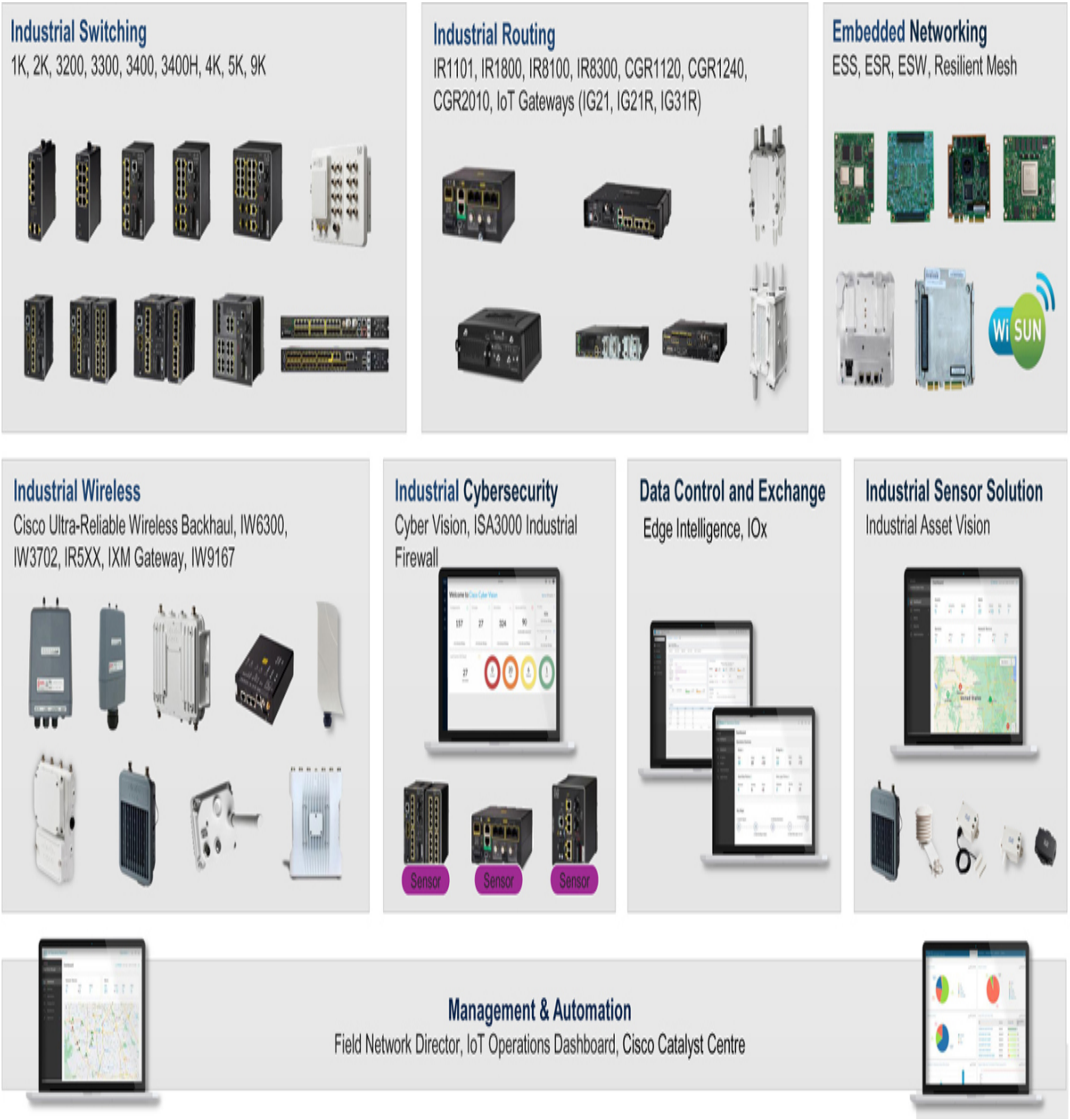

Industrial Switching
1K, 2K, 3200, 3300, 3400, 3400H, 4K, 5K, 9K
Industrial Routing
IR1101, IR1800, IR8100, IR8300, CGR1120, CGR1240, CGR2010, IoT Gateways (IG21, IG21R, IG31R)
Embedded Networking
ESS, ESR, ESW, Resilient Mesh
Wi SUN
Industrial Wireless
Cisco Ultra-Reliable Wireless Backhaul, IW6300, IW3702, IR5XX, IXM Gateway, IW9167
Industrial Cybersecurity
Cyber Vision, ISA3000 Industrial Firewall
Sensor
Sensor
Sensor
Data Control and Exchange
Edge Intelligence, IOx
Industrial Sensor Solution
Industrial Asset Vision
Management & Automation
Field Network Director, IoT Operations Dashboard, Cisco Catalyst Centre

Figure 21-1 *Cisco Industrial IoT Products*

A typical industrial plant follows the Purdue Model for industrial control systems (ICS) security. The Purdue Model for ICS security serves as an essential framework for segmenting ICS networks. Its primary goal is to safeguard operational technology from malware and various other attacks. It enhances the security and manageability of the network by dividing it into distinct levels with well-defined functions. The zero trust model can be regarded as integrated into the Purdue framework and operating on the principle that no segment of the network should be automatically trusted. Figure 21-2 shows the Purdue Model.

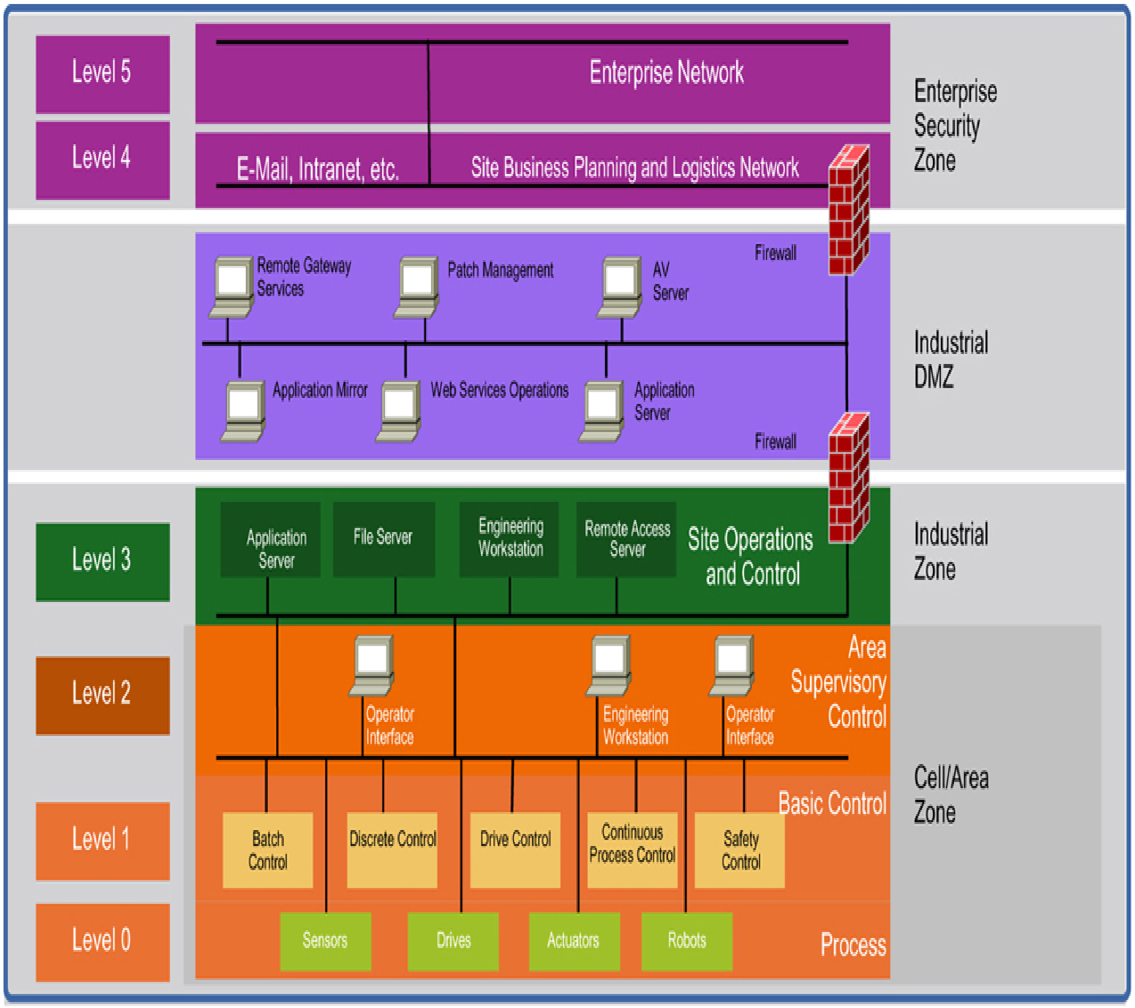


Figure 21-2 *Purdue Model*

The Purdue Model divides the entire industrial network into six levels (0 to 5). The top two levels—Level 5 and Level 4—fall under the enterprise function of any industrial plant. Every plant will have a basic IT setup to provide services such as email, order processing, Internet, and logistics. A network that caters to these and similar IT needs falls under these top two levels.

Levels 0 to 3 form the main OT part of a plant and are segregated from Levels 4 and 5 by the industrial DMZ (IDMZ) block. This block, as you will learn later in this chapter, helps create the macro-level segmentation for our zero trust architecture (ZTA) for a manufacturing plant. Let's look more closely at Levels 0 to 3 for the OT block:

- **Level 0:** This level comprises components such as sensors, actuators, robots, and other physical devices.
- **Level 1:** This level comprises programmable logic controllers (PLCs) and remote terminal units (RTUs). You will see a typical requirement to either block or allow communication between a set of PLCs within an OT plant. These PLCs are responsible for executing control algorithms, and they send commands to devices in Level 0 and receive inputs from them.
- **Level 2:** This level is known as supervisory control and has components like human-machine interfaces (HMIs) and SCADA systems. These are computers or tablets that allow specific software to run so that human operators can feed in specific commands. These commands or procedures are then sent to the PLCs.
- **Level 3:** This level primarily includes manufacturing execution systems (MES). Its primary function is to coordinate the production workflows and create schedules. In networking terms, you will see servers in this layer running specific functions and protocols for the OT network.

Levels 0 to 2 are collectively known as *cell/area zones*. In a typical industrial network, you will see multiple cell/area zones. As an example, in a milk processing plant, one cell might process and bottle milk while another cell will make ice creams. You might have many other cells for products like butter or cheese. Usually, coordination or control is required between machines (PLCs) running in different cells. For our example of a milk processing plant, milk is the main ingredient for all other products, so it means that the PLC controlling the milk supply needs to communicate with PLCs in butter, cheese, and ice cream. You might think of this as a vague example, but typical business requirements from a manufacturing customer will be along similar lines.

One of my customers expressed a business problem, saying that milk was not getting distributed to the ice cream plant, and asked for help to solve this issue. My team later analyzed and found that, due to a lack of segmentation in the network, a broadcast storm in the network was hindering the PLC-to-PLC communication. Figure 21-3 shows how multiple cell/area zones can be part of the same manufacturing plant.

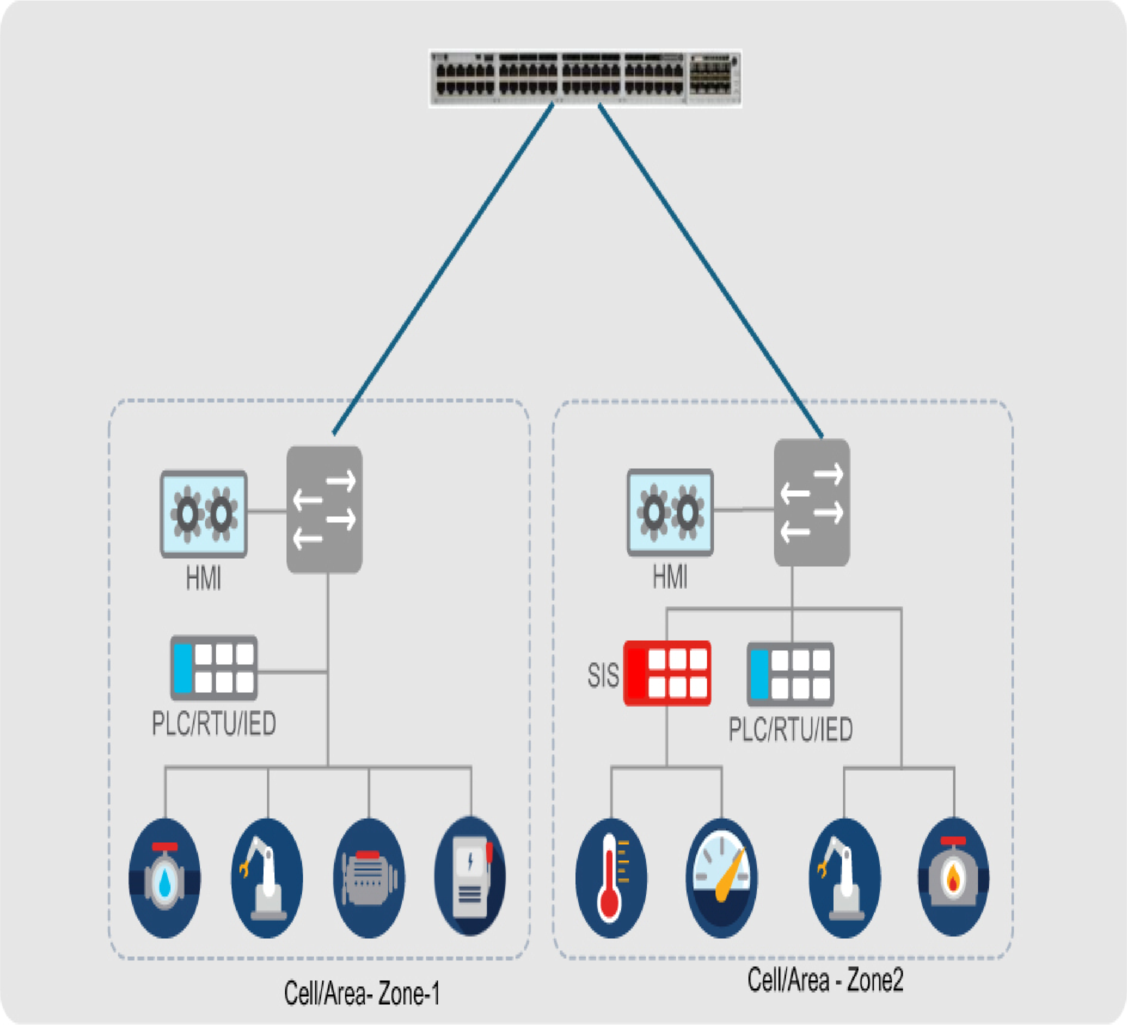


Figure 21-3 *Multiple Cell/Area Zones Within a Plant*

Before looking into the security aspects of industrial networks, you must familiarize yourself with *Converged Plantwide Ethernet (CPwE)* architecture. This reference architecture provides guidelines for designing and deploying resilient, secure, and scalable industrial networks. Developed collaboratively by Cisco and Rockwell Automation, CPwE aims to integrate industrial automation and control systems with enterprise networks using standard Ethernet and IP networking technologies. It is based on the Purdue Model discussed previously. You can think of CPwE as a practical way of implementing the Purdue Model that supports defense-in-depth strategies with multiple layers of security control. Macrosegmentation that is built into the CPwE reference architecture uses firewalls and intrusion detection systems/intrusion prevention systems (IDS/IPS) to segregate IT from OT areas and different OT areas from each other.

Figure 21-4 shows the Cisco CPwE architecture framework. Notice the different networking devices positioned at different levels. This reference architecture is aligned with the Purdue Model. Also, note that devices on Levels 3 to 5 are typical enterprise devices, whereas the devices in Levels 0 to 3 are ruggedized industrial routers, switches, and wireless access points.

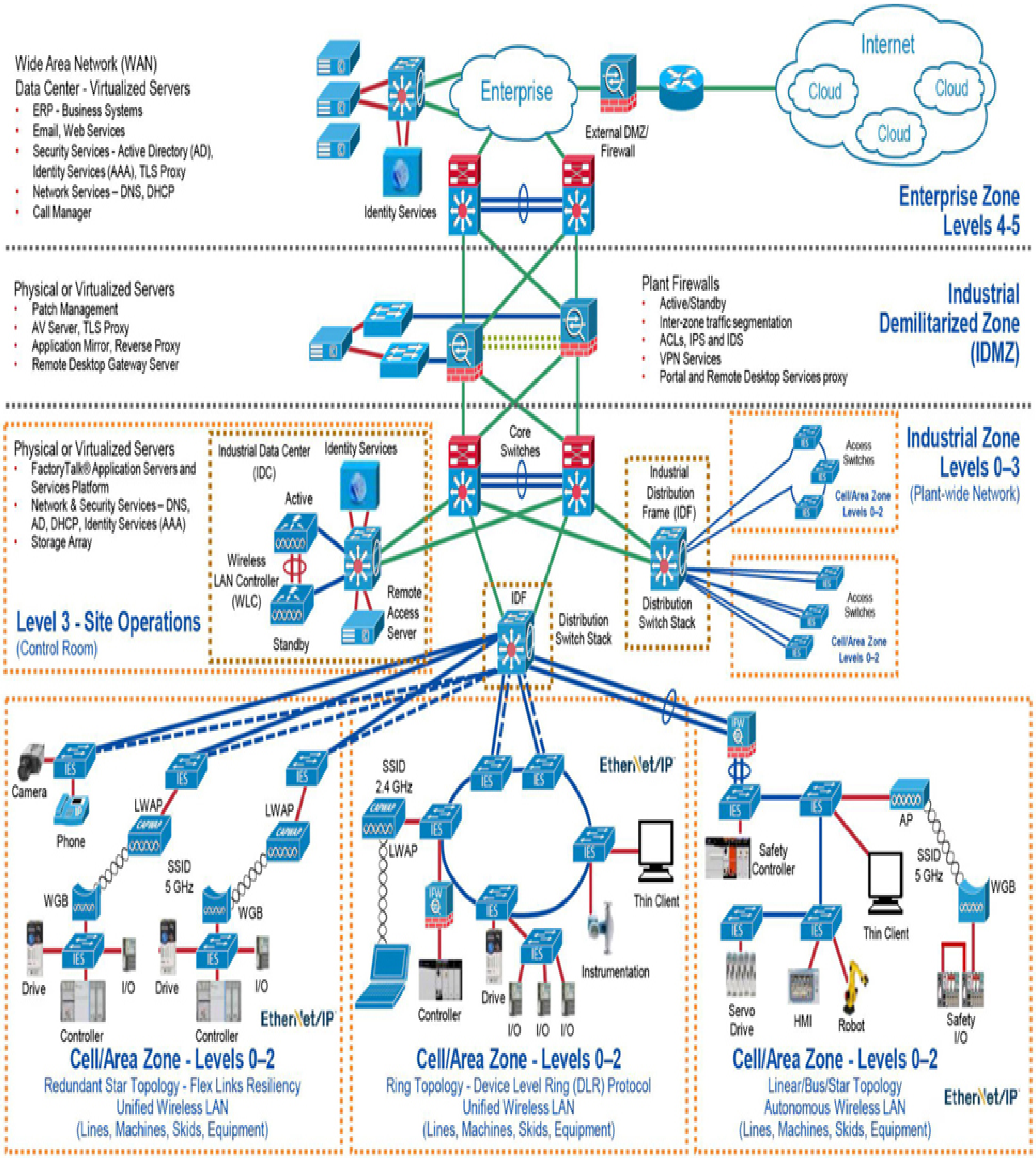

Wide Area Network (WAN)
Data Center - Virtualized Servers
ERP - Business Systems
Email, Web Services
Security Services - Active Directory (AD), Identity Services (AAA), TLS Proxy
Network Services – DNS, DHCP
Call Manager
Identity Services
Enterprise
External DMZ/ Firewall
Internet
Cloud
Cloud
Cloud
Enterprise Zone Levels 4-5
Physical or Virtualized Servers
Patch Management
AV Server, TLS Proxy
Application Mirror, Reverse Proxy
Remote Desktop Gateway Server
Plant Firewalls
Active/Standby
Inter-zone traffic segmentation
ACLs, IPS and IDS
VPN Services
Portal and Remote Desktop Services proxy
Industrial Demilitarized Zone (IDMZ)
Physical or Virtualized Servers
FactoryTalk® Application Servers and Services Platform
Network & Security Services – DNS, AD, DHCP, Identity Services (AAA)
Storage Array
Industrial Data Center (IDC)
Identity Services
Active
Wireless LAN Controller (WLC)
Standby
Remote Access Server
Core Switches
Industrial Distribution Frame (IDF)
Distribution Switch Stack
IDF
Distribution Switch Stack
Access Switches
Cell/Area Zone Levels 0-2
Access Switches
Cell/Area Zone Levels 0-2
Industrial Zone Levels 0–3 (Plant-wide Network)
Level 3 - Site Operations (Control Room)
IES
Camera
Phone
LWAP
SSID 5 GHz
LWAP
WGB
WGB
Drive
I/O
Drive
I/O
Controller
Controller
EtherNet/IP
Cell/Area Zone - Levels 0–2
Redundant Star Topology - Flex Links Resiliency
Unified Wireless LAN
(Lines, Machines, Skids, Equipment)
SSID 2.4 GHz
LWAP
EtherNet/IP
Thin Client
IFW
Controller
Drive
I/O
I/O
I/O
Instrumentation
Cell/Area Zone - Levels 0–2
Ring Topology - Device Level Ring (DLR) Protocol
Unified Wireless LAN
(Lines, Machines, Skids, Equipment)
IFW
Safety Controller
AP
SSID 5 GHz
WGB
Thin Client
Servo Drive
HMI
Robot
Safety I/O
Cell/Area Zone - Levels 0–2
Linear/Bus/Star Topology
Autonomous Wireless LAN
(Lines, Machines, Skids, Equipment)
EtherNet/IP

Figure 21-4 *Cisco CPwE Framework with Typical Device Roles*

Pillars of ZTNA for Industrial Plant Networks

It is important to understand the main functions and components of each level of the Purdue Model because segmentation policies will need to be created either between the devices within the same cell/area zone or between the cell/area zones. This brings us to the important question of how we can plan zero trust in industrial networks. The following are the key pillars of zero trust network access (ZTNA) in the industrial plant:

1. Build a security foundation using macrosegmentation
2. Gain visibility with the network as a sensor
3. Create smaller trust zones using microsegmentation
4. Use correlation and automation to contain threats

Security Foundation with Firewalls

Macrosegmentation is the process of dividing a network into broad segments based on functional or operational criteria. In an industrial setting, these segments could include production lines, quality control, inventory management, and administrative functions. This segmentation helps to isolate different parts of the network, minimizing the impact of security breaches and improving overall network performance. The approach involves dividing the network into large, distinct virtual networks usually employing the concept of industrial DMZ.

In reference to CPwE, you have to create IDMZ with firewalls between the IT and OT levels. This creates a three-zone model:

- **Enterprise (IT) Zone:** This is the zone where corporate IT systems, business applications (e.g., ERP, MES), and user devices reside.
- **Industrial DMZ (IDMZ):** This is the secure buffer zone between the enterprise IT network and the operational OT network.
- **Industrial (OT) Zone:** This zone includes the plant floor, SCADA systems, PLCs, HMIs, and other operational devices. By default, any direct communication between IT and OT should be disallowed.

Any application required by the OT network must be serviced from the IDMZ. Typical applications like file transfer, remote desktops, and software updates will be hosted in the IDMZ. In summary, any north-south traffic has to go through the firewalls. In this scenario, we can use the enterprise firewalls. Industrial firewalls like Cisco ISA3000 may not be required because most traffic between IT and OT is TCP/UDP-based, such as order processing, email access, or limited telemetry to the cloud. Figure 21-5 shows IT-OT segmentation using firewalls.

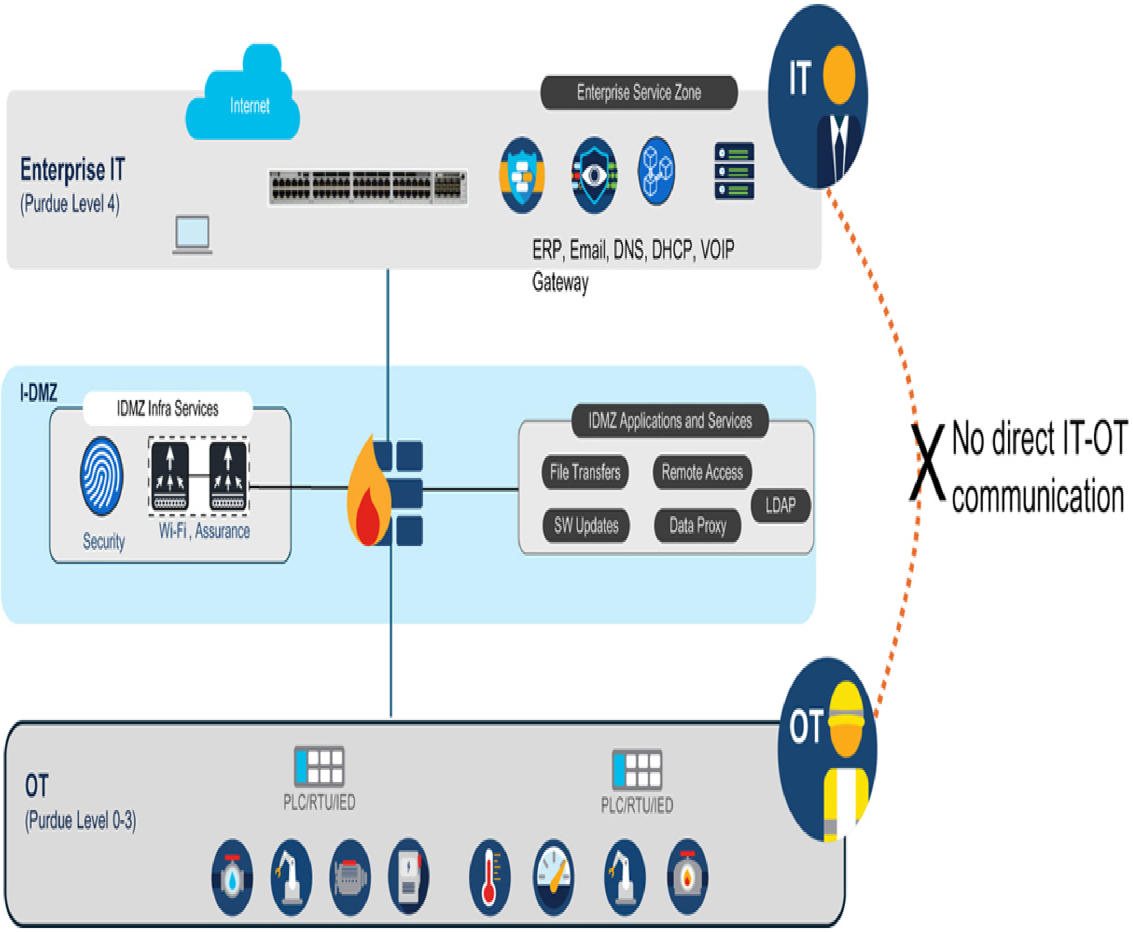


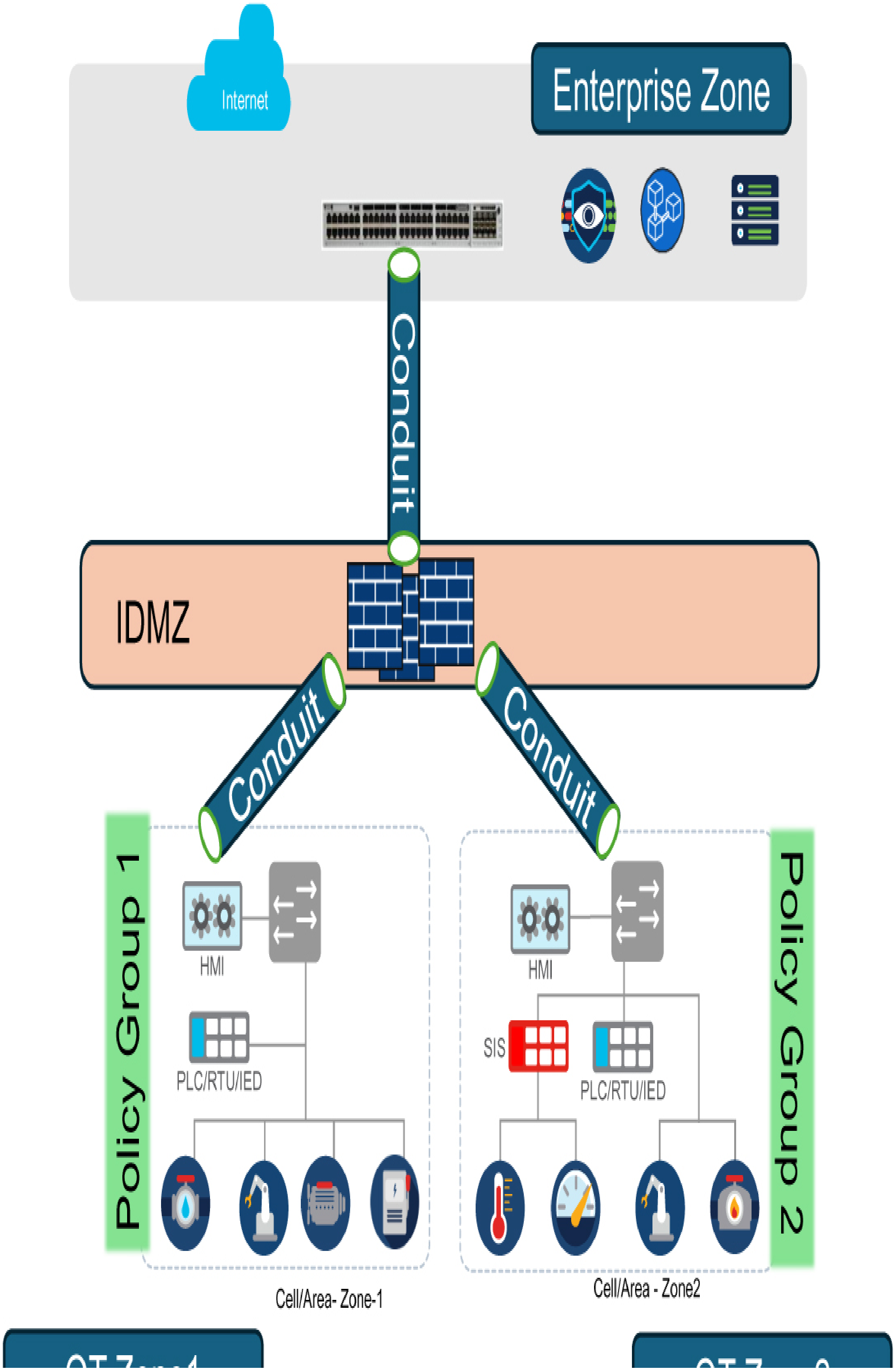
Figure 21-5 *OT Communication via IDMZ*

In some cases, there might be a requirement to control the communication between two cell/area zones. The purpose is to completely block the communication between two plants with no interdependencies between them. In such cases, firewall-based isolation between plants can be created. I have also seen some customers deploying quality assurance (QA) systems such as camera-based defect detection systems isolated from the main production OT environment. This becomes important when such systems continuously send telemetry or interact with cloud-based systems outside OT boundaries. Such macro-level isolation prevents the main production system, in rare cases of QA systems getting compromised. Remember that production continuity is the main requirement for an industrial establishment.

IEC 62443 is a series of standards that focuses on cybersecurity for operational technology. It recommends dividing the OT area in zones and conduits. The intent is to identify and group OT assets that share common security characteristics:

- **Zones:** Collections of entities that represent the partitioning of a system under consideration based on the functional, logical, and physical (location) relationships that share common security requirements.
- **Conduit:** Physical or logical grouping of communication channels, intermittent or permanent, connecting zones with other zone or with the outside network that share the common security requirements.

This is different from an enterprise segmentation strategy where segmentation is based on device types and not on zones. A common but not mandatory requirement for OT design is to allow device communication between zones but to restrict inter-zone communication. Figure 21-6 shows the CPwE as IEC 62443.



Internet
Enterprise Zone
Conduit
IDMZ
Conduit
Conduit
Policy Group 1
HMI
PLC/RTU/IED
Cell/Area- Zone-1
HMI
SIS
PLC/RTU/IED
Policy Group 2
Cell/Area - Zone2

Figure 21-6 *IEC 62443 Zones and Conduits*

Visibility with the Network as a Sensor

Visibility into any network is critical to understanding the state of the network. Industrial OT networks use special protocols for communication, such as PROFINET and MODBUS. These protocols are used for communication between different OT components like SCADA and PLCs and IO devices like motors and actuators. Typical enterprise visibility tools will not be able to understand these protocols and communication between industrial components. OT networks need to deploy a visibility system specific to OT needs. One such observability tool is Cisco Cyber Vision (CV). This tool monitors OT events, identifies network problems, and helps you troubleshoot issues faster. More importantly, it allows you to segment the OT network with integration with Cisco ISE.

The Cisco Cyber Vision architecture has two components: Sensors and Cyber Vision Center.

1. **Sensors:** The sensor component is responsible for collecting the data from the OT network. These sensors perform deep packet inspection and are responsible for creating metadata, which is then shared with the Cyber Vision Center. These sensors can be deployed in various forms such as the following:

   a. **Network Sensors:** These sensors are integrated into the Cisco network devices like industrial switches and routers: Cisco IE3400, Cisco IE5000, Cisco Catalyst 9300, IR1101, and so on.

   b. **Hardware Sensors:** This is a dedicated hardware sensor appliance Cisco IC3000 Industrial Compute Gateway. It is useful in scenarios where OT architecture is not using Cisco switches and wishes to use Cisco Cyber Vision. A SPAN session for target OT traffic is then created to Cisco IC3000, where it is processed, and metadata is then generated.

2. **Cyber Vision Center:** This is the central management console. It aggregates data from all the sensors, processes that data, and

provides a comprehensive view of the network security posture. It can be deployed as a virtual server on ESXI and is also available as a physical server appliance using Cisco Unified Computing System (UCS). Cloud installation on Microsoft Azure and Amazon Web Services (AWS) is also supported. It is responsible for

a. **Data Analysis and Correlation:** Analyzes network traffic and correlates to detect anomalies and potential threats.

b. **Dashboard and Reporting:** Offers a user-friendly dashboard to visualize network activity, alerts, and reports.

c. **Policy Management:** Allows administrators to define and manage security policies for the network.

You can also merge the data from multiple Cyber Vision Centers into a single dashboard via the Global Cyber Vision Center to provide a single view across all OT plants. Global Center view is typically deployed in a distributed factory environment where plants are in different locations. This is an optional component.

Sensors share the lightweight metadata only with the Cyber Vision Center for further analysis. Figure 21-7 shows the overall architecture of Cyber Vision in the Purdue Model. You will note that cell/area zone 1 switch supports the sensor on the Cisco IE switch. Cell/area zone 2 uses a third-party switch where the CV sensor cannot be deployed directly. In this case, a SPAN session is created with Cisco IC3000 acting as a CV sensor. Sensors are responsible for doing the deep packet analysis. In both cases, raw data is not shared with Cyber Vision Center; instead, lightweight metadata is sent for further analysis.

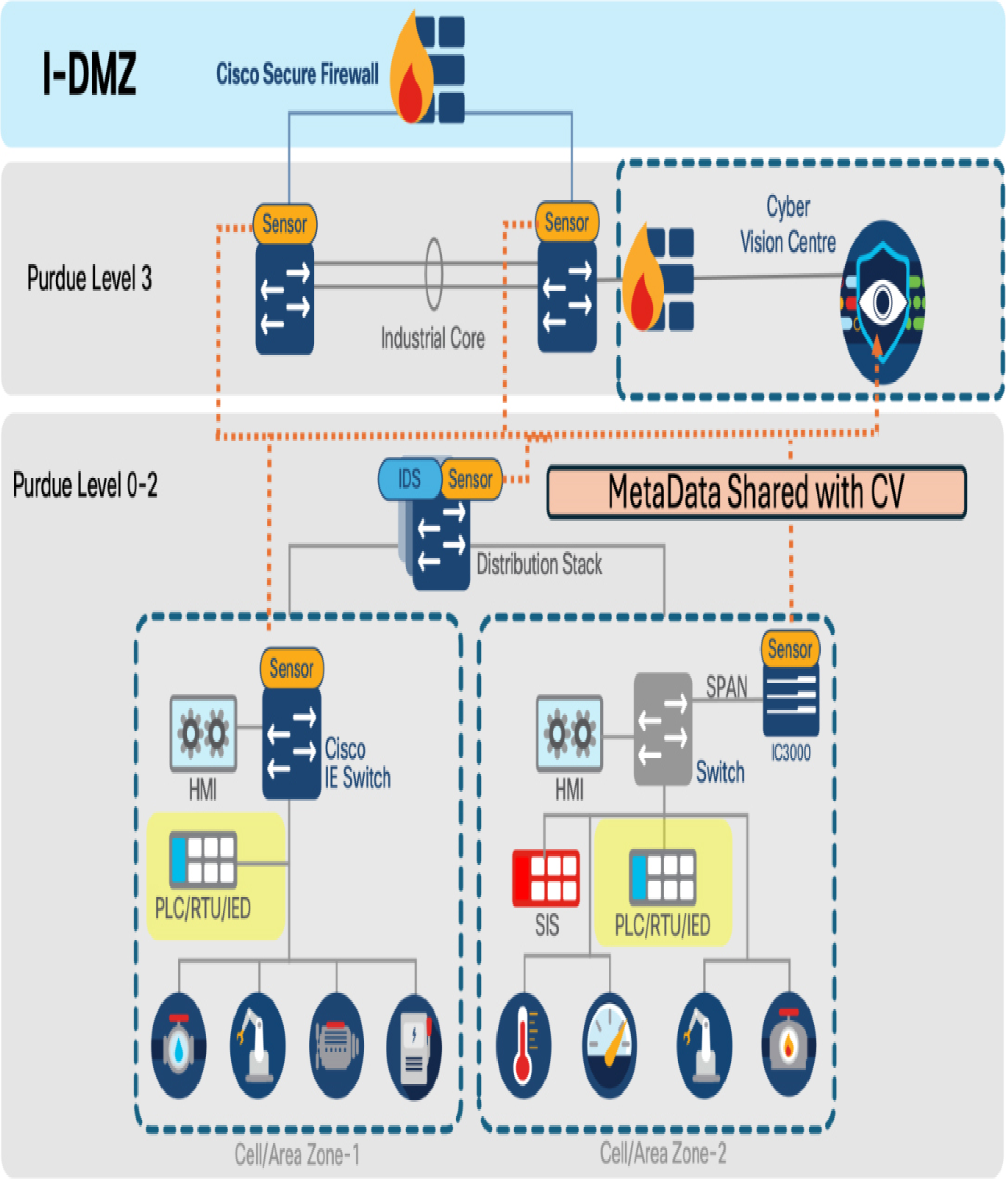


Figure 21-7 *Cisco Cyber Vision Placement in Purdue Model*

The metadata collected from the sensors is continuously analyzed by the Cyber Vision Center, allowing near real-time visibility into the security threats of the OT environment. Cyber Vision Center provides a detailed list of vulnerabilities found in the OT environment. With sensors embedded into the network devices, the network itself acts as a sensor. This allows information to be captured at the edge of the network. It is also important to note that only metadata is shared with the center. This reduces the amount of traffic and thereby reduces congestion for the critical OT traffic.

Cyber Vision assigns the tags to the components or flows when it detects any behavior, protocol, or critical command execution. These tags are used to classify the components and flows. As an example, when a program is downloaded on a PLC, it is tagged as red, with the Program Download tag. Similarly, all OT communications are tagged. In another example, when a PLC is detected using protocol Siemens S7, CV assigns the tags S7 and PLC. Cyber Vision also uses the concept of risk score, which is an indicator of good health and criticality level and ranges from 0 to 100. High risk is marked in red with scores in the range 0 to 100, medium risk is marked in orange with score values between 40 and 69, and low-risk level is marked in green with scores in the range of 0 to 39. OT administrators can easily identify the critical devices using the risk score and can take immediate action to solve the identified risk.

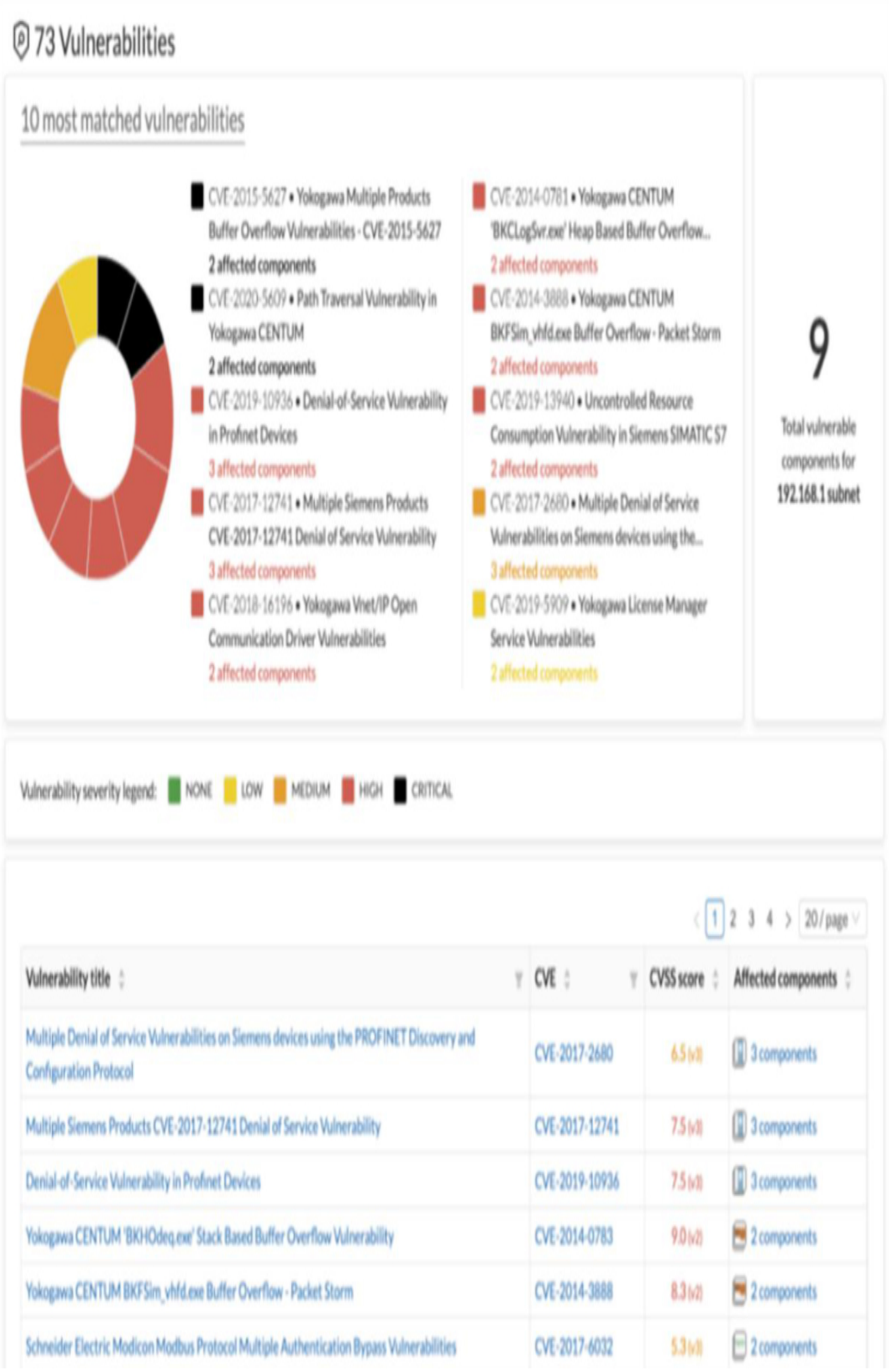
The risk score is calculated using the following formula:

Risk = Impact × Likelihood

The impact depends on the position of the device in the network and what impact it can have on the overall network. For example, if the device is an endpoint like a simple IO device connected to a single machine or a SCADA system that controls the entire plant. OT admins can manually assign the devices into different groups based on the criticality of the role they play in the plant.

The likelihood defines the probability of the device being compromised. It uses various parameters, such as what kind of protocol the device is using (e.g., SSH vs. Telnet). How often do the devices communicate and to which devices? What are the current device vulnerabilities?

Cyber Vision continuously evaluates the OT devices for various vulnerabilities. This is done based on the rules stored in the internal knowledge database (KDB) of Cyber Vision Center. This KDB is sourced from several Computer Emergency Response Teams (CERTs), and OT manufacturers like Schneider and Siemens. This KDB is available as a file that the security admin needs to download from the Cisco website, which is then uploaded to the Cyber Vision Center. It is important that OT admins keep the knowledge DC on Cisco Cyber Vision updated. Figure 21-8 shows the Vulnerabilities dashboard of CV.



73 Vulnerabilities

10 most matched vulnerabilities

⬛ CVE-2015-5627 • Yokogawa Multiple Products Buffer Overflow Vulnerabilities - CVE-2015-5627
2 affected components

⬛ CVE-2020-5609 • Path Traversal Vulnerability in Yokogawa CENTUM
2 affected components

🟥 CVE-2019-10936 • Denial-of-Service Vulnerability in Profinet Devices
3 affected components

🟥 CVE-2017-12741 • Multiple Siemens Products CVE-2017-12741 Denial of Service Vulnerability
3 affected components

🟥 CVE-2018-16196 • Yokogawa Vnet/IP Open Communication Driver Vulnerabilities
2 affected components

🟥 CVE-2014-0781 • Yokogawa CENTUM 'BKCLogSvr.exe' Heap Based Buffer Overflow...
2 affected components

🟥 CVE-2014-3888 • Yokogawa CENTUM BKFSim_vhfd.exe Buffer Overflow - Packet Storm
2 affected components

🟥 CVE-2019-13940 • Uncontrolled Resource Consumption Vulnerability in Siemens SIMATIC S7
2 affected components

🟧 CVE-2017-2680 • Multiple Denial of Service Vulnerabilities on Siemens devices using the...
3 affected components

🟨 CVE-2019-5909 • Yokogawa License Manager Service Vulnerabilities
2 affected components

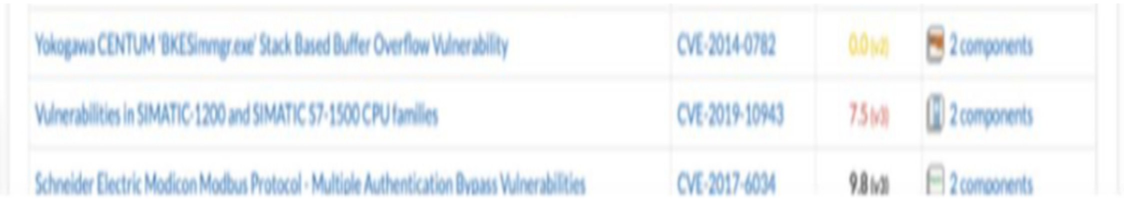
9

Total vulnerable components for **192.168.1 subnet**

Vulnerability severity legend: 🟩 NONE 🟨 LOW 🟧 MEDIUM 🟥 HIGH ⬛ CRITICAL

‹ 1 2 3 4 › 20 / page

Vulnerability title	CVE	CVSS score	Affected components
Multiple Denial of Service Vulnerabilities on Siemens devices using the PROFINET Discovery and Configuration Protocol	CVE-2017-2680	6.5 (v3)	3 components
Multiple Siemens Products CVE-2017-12741 Denial of Service Vulnerability	CVE-2017-12741	7.5 (v3)	3 components
Denial-of-Service Vulnerability in Profinet Devices	CVE-2019-10936	7.5 (v3)	3 components
Yokogawa CENTUM 'BKHOdeq.exe' Stack Based Buffer Overflow Vulnerability	CVE-2014-0783	9.0 (v2)	2 components
Yokogawa CENTUM BKFSim_vhfd.exe Buffer Overflow - Packet Storm	CVE-2014-3888	8.3 (v2)	2 components
Schneider Electric Modicon Modbus Protocol Multiple Authentication Bypass Vulnerabilities	CVE-2017-6032	5.3 (v3)	2 components



Yokogawa CENTUM 'BKESimmgr.exe' Stack Based Buffer Overflow Vulnerability	CVE-2014-0782	0.0 (v2)	2 components
Vulnerabilities in SIMATIC-1200 and SIMATIC S7-1500 CPU families	CVE-2019-10943	7.5 (v3)	2 components
Schneider Electric Modicon Modbus Protocol - Multiple Authentication Bypass Vulnerabilities	CVE-2017-6034	9.8 (v3)	2 components

Figure 21-8 *Vulnerabilities Dashboard on Cisco Cyber Vision Center*

Cyber Vision also allows active discovery of the OT components to help build the network inventory and create a list of custom attributes that is then shared with AAA servers like Cisco ISE to be used to create policy constructs. This active discovery is done in two modes:

- **Broadcast:** The sensors send the broadcast packets (industrial protocols like EtherNet/IP, PROFINET, SiemensS7) targeting all the devices in the subnet. The devices that support these protocols will then respond.
- **Unicast:** The sensors send the unicast discovery packets of industrial protocols like BACnet, DNP3, and MODBUS to gather information about the devices.

Active discovery also aids in identifying rogue devices and unauthorized devices in the network. It is common for OT vendors to add additional devices into the network as and when the network grows.

With Cisco Cyber Vision, you can look for insights into the OT network protocols, vulnerabilities, device inventory, and associated risks. This is a critical step in creating dynamic microsegmentation policies, which are described in the next section.

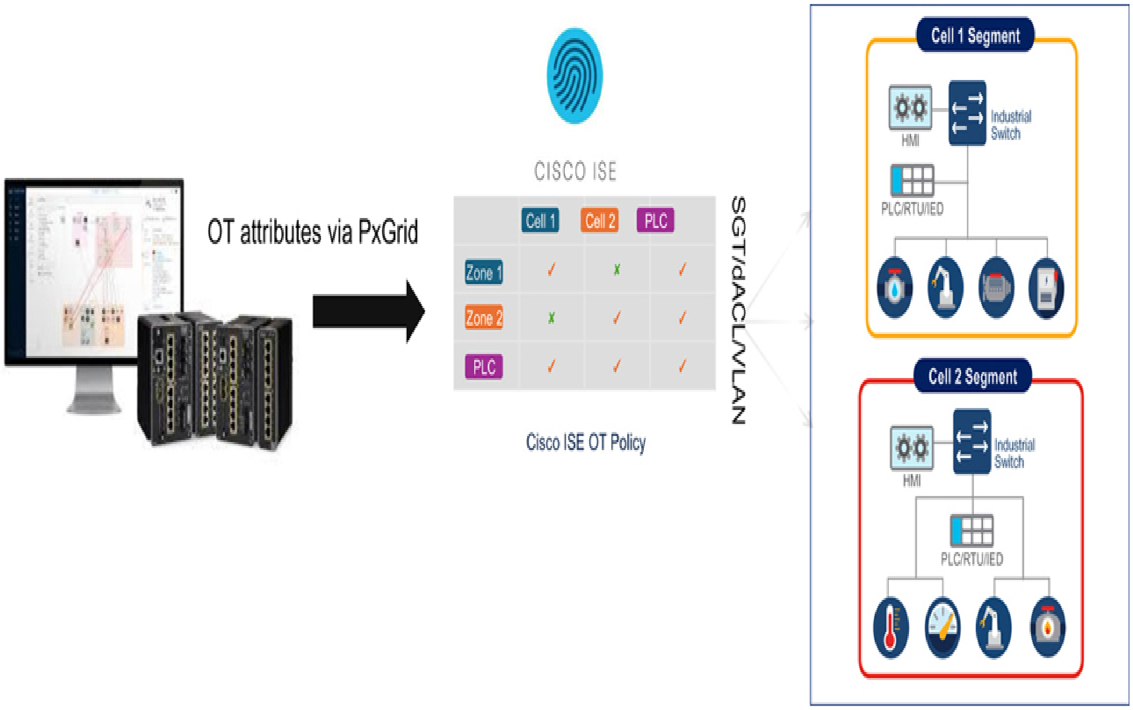
Creating Granular Trust Zones Using Microsegmentation

Microsegmentation in an industrial OT environment refers to the segmentation between or within a cell/area zone. This falls under Levels 0 to 3 of the Purdue Model. To create such segmentation, you will need to classify the assets, put them in specific groups, and then apply policies. Asset visibility and attributes to classify the industrial assets can be fetched from an OT visibility platform like Cisco Cyber Vision. In the previous section, you learned how Cyber Vision can do asset discovery using active and passive methods. This process also discovers various attributes of the OT devices such as assetID, assetName, assetVendor, and assetProtocol. These attributes are then shared with a AAA server like Cisco ISE using the custom policies that can be created. These attributes are shared using PxGRID. Figure 21-9 shows how OT attributes can be used by Cisco ISE to create policy constructs. Administrators can then create custom rules and policies to restrict communication within or across the cell/area zone. Based on the segmentation mechanism supported on the OT infrastructure, the following actions can be taken:

- **Dynamic VLAN Assignment:** This is a traditional method of segmentation but still common in industrial environments. Based on the asset type, group, and function, you can dynamically group devices within a cell/area in one or more VLANs based on the business requirements.

- **Dynamic ACL:** This is the least used mechanism, where a downloadable ACL is used with a wired or wireless network to restrict the communication between groups of users or devices. The ACL needs to be updated every time the OT device IP changes.

- **Security Group Tags (SGTs):** Segmentation using SGTs is more flexible and allows intent-based network segmentation. This also makes network operations and troubleshooting simpler because policy contracts are simpler and the operations team does not have to worry about IP-based ACLs.



	Cell 1	Cell 2	PLC
Zone 1	✓	✗	✓
Zone 2	✗	✓	✓
PLC	✓	✓	✓

Figure 21-9 *OT Attributes Shared with Cisco ISE Using PxGrid*

Cisco Software-Defined Access (SD-Access) represents a sophisticated network solution that facilitates the simplification and automation of the design, deployment, and management processes associated with enterprise networks. This solution leverages the principles of software-defined networking (SDN) to construct a cohesive fabric that supports both wired and wireless connectivity. Within the policy plane of Cisco SD-Access, security group tags (SGTs) serve a vital function in the implementation of identity-driven policies throughout the network. SGTs function as metadata that are allocated to users, devices, or applications predicated on their identity, role, or group affiliation. These tags empower the policy plane to enforce role-based access control (RBAC) and micro-segmentation in a dynamic and consistent manner.

It is worthwhile to go deeper into the SGT-based segmentation because it is used in both traditional (no Cisco SD-Access) and Cisco SD-Access[nd]based OT policy definitions. SDA architecture and policy constructs are discussed later in this chapter. The following steps are required to deploy SGT-based OT network segmentation

1. **Define Security Groups:**

   a. **Classify Assets:** Group devices and systems based on their role, function, and security needs (e.g., PLCs, SCADA, HMI). Cisco Cyber Vision could be used for device inventory.

   b. **Create SGTs:** Define SGT tags for each category, ensuring that each group has a unique tag.

2. **Assign SGTs:**

   a. **Dynamic Assignment:** Cisco ISE policy can be created to assign the SGTs based on authentication and authorization.

   b. **Static Assignment:** SGTs can also be statically assigned to the group of devices. This capability will be useful for older generations of devices like PLC or other industrial systems that are not designed to support AAA-based authentication and authorization. In some cases, OT networks are not very dynamic and may not benefit from dynamic SGT process examples, where the entire subnet needs to be mapped to a single SGT.

3. **Ensure SGT Propagation:**

   a. Network devices like OT routers and switches need to be configured to propagate SGT across domains.

   b. If there are non-TrustSec devices in the OT network, the SGT exchange protocol needs to be configured.

4. **Define Policy Contracts:**

   a. Create policy definition on Cisco ISE to control traffic flow between different SGT groups.

   b. Configure the policy enforcement points.

Let’s examine this topic in more detail with some examples and use cases.

Use Case 1

A manufacturing plant operation is divided into two cells/areas: cell 1 and cell 2. Because both operations are mutually exclusive, the business wants to restrict all communication between cell/area zone 1 and cell/area zone 2.

In this case, the entire subnet serving a specific cell/area zone could be classified under one SGT. In this way, you will have a subnet from cell/area zone 1 mapped with SGT Cell1 and a subnet from cell/area zone 2 mapped with SGT Cell2. Policy contracts can then be created to disallow communication between SGT Cell1 and SGT Cell2. The best place for enforcing such a policy will be at the distribution switch, as shown in Figure 21-10.

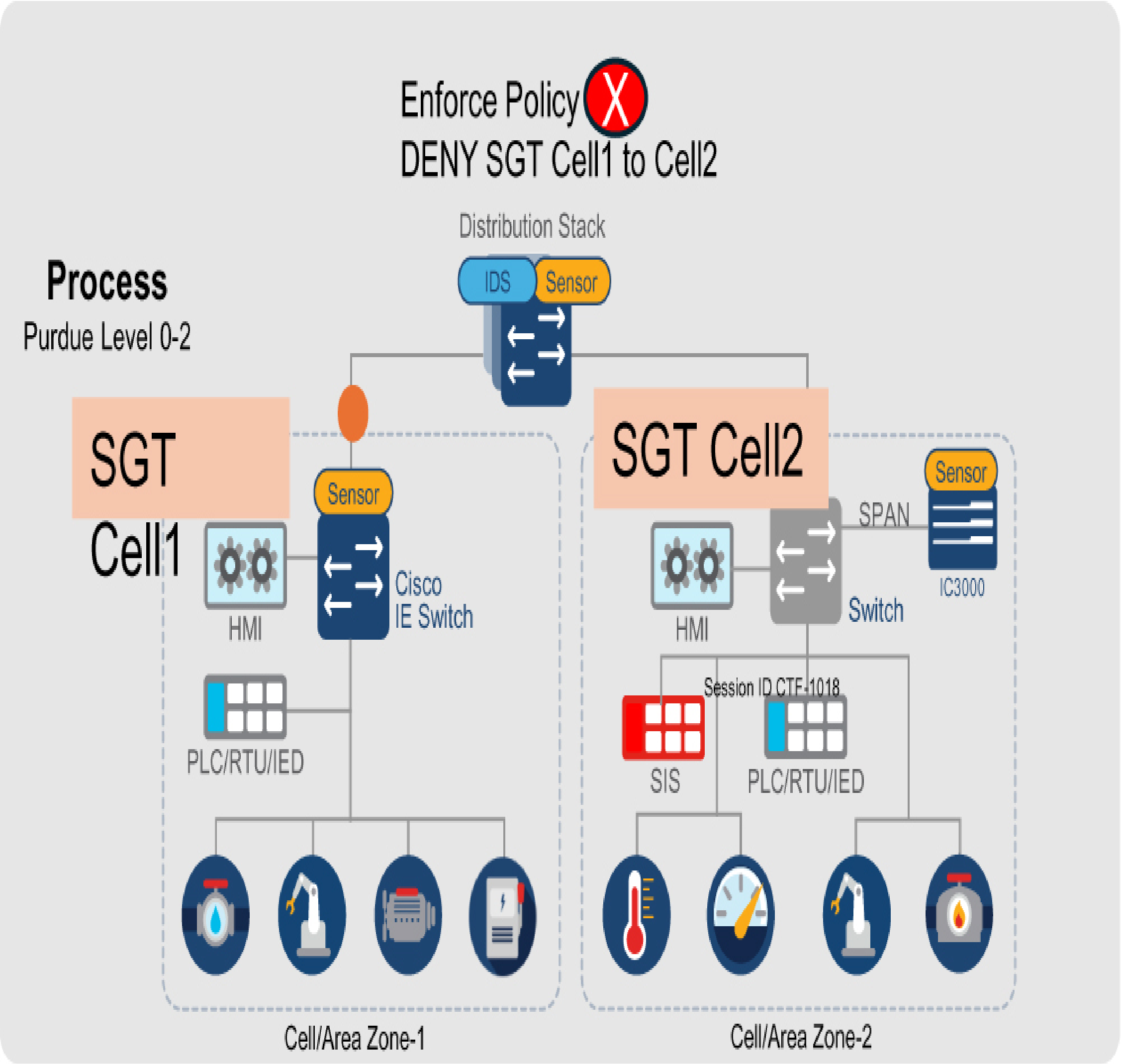


Figure 21-10 *SGT-Based Microsegmentation Between OT Cells*

Use Case 2

In this use case, the OT team wants to restrict all the communication between Zone 1 and Zone 2 except specific interlocking PLCs.

In this scenario, instead of assigning an SGT to the entire subnet, we can assign the SGT to a specific group of devices—in this case, PLCs. Each cell/area zone device like PLC can be grouped in a separate SGT while SCADA and HMIs could be another SGT group and the remaining devices could be under a third SGT. Policy contracts can then be created to allow communication between specific SGT groups. Alternatively, if communication between specific PLCs is to be blocked, this requirement could also be achieved by creating new contracts specifically to block communication between SGTs.

East-west communication traffic flow happening across the cell/area zone is most common in manufacturing plants. Because different manufacturing processes consume inputs from each other, most of the traffic flows east-west. In practical scenarios, you will see a lot of requirements to restrict or allow certain communication in east-west fashion. The first step in such cases is to identify the traffic patterns in terms of devices, industrial protocols, and so on. Cyber Vision can help here to collect the traffic patterns between devices. With visibility into OT network flows and protocols, network segmentation policies at the micro level can be fine-tuned.

North-south communication is the traffic flow between different layers of the OT network as defined in the Purdue Model. An example would be communication from the engineering workstation from Level 3 to a controller, which is Level 2. Specific policies need to be defined to allow only authorized communication.

Correlation and Automation to Contain Threats

As we covered previously, industrial security uses different components like firewalls, AAA servers, observability platforms like Cyber Vision, and endpoint security.

It is critical to have a well-defined response strategy for different threats. Manually coordinating between teams and investigating incidents on multiple consoles are time-consuming. There is a need for systems that consume information from multiple sources, correlate the information, visualize the information in a single place, and also take well-defined action needed for next-generation OT environments.

For the Cisco OT solution, this could be achieved by integrating Cisco Cyber Vision with Cisco Extended Detection and Response (XDR). Cisco XDR is a comprehensive security solution designed to improve threat detection, investigation, and response orchestration. It provides a cohesive and automated security framework. In the context of industrial security environments, Cisco XDR and OT work together as follows:

1. Cisco Cyber Vision provides visibility into OT assets like device types, firmware versions, and communication patterns to Cisco XDR, enhancing its understanding of the OT environment.

2. Anomalies and threat data from Cisco Cyber Vision are sent to Cisco XDR, where they are correlated with data from other OT and IT security tools like firewalls and IDS/IPS. This creates a holistic view of the threat landscape across IT and OT networks.

3. Playbooks are then defined in Cisco XDR to automate the threat response actions. These playbooks contain trigger actions to be sent to various component parts of OT security like firewalls and AAA servers.

4. Threats from Cyber Vision can also be reviewed and converted to incidents via XDR. OT administrators will usually create a baseline for their OT network operations. Any change in the baseline is recorded with details like new message/communication frames. This can then be converted to incidents, and an auto action can be triggered when authorized by OT admins.

Let’s examine the power of automation and correlation more with an example. Imagine a hypothetical scenario where Cisco Cyber Vision is continuously monitoring the OT environment. This environment consists of a number of PLCs, SCADA servers, endpoints, HMIs, and so on. Admins have already done baselining for this OT environment, so when an unusual communication pattern is detected from PLC, indicating potential malware, this anomaly is sent to Cisco XDR. This anomaly is then analyzed in depth by Cisco XDR by correlating with security data from other components, identifying malware as part of a known attack pattern. XDR determines the scope of the infection and potential impacts. Cisco XDR then triggers the automated response playbook to isolate the PLC and update the firewall rules to block the malicious traffic from the malware. The event is then logged as an incident for security teams to further identify the root cause and implement additional security measures. Insights from such incidents are then used to update the Cisco XDR playbooks and OT network security policies resulting in continuous improvements.

At the time of writing this chapter, Cisco is actively working on integrating Cisco XDR with Splunk Enterprise and OT security. This will bring the advanced AI analytics capability and further strengthen the OT/IT security posture.

USB and external hard devices are still common methods to bring in the OT firmware and other software upgrade elements. However, many times, these devices carry malware infections and other network viruses, resulting in network breaches. A common strategy is to dedicate a computer to OT with limited or no connectivity to the network and install with advanced malware and endpoint security software to scan such devices. This process of using a dedicated device to test the inbound removable media is known as a *sheep-dip*.

Apart from a sheep-dip, it is highly recommended to install the endpoint security solution on all applicable workstations. However, it may not be possible to install such security systems on constrained devices like HMIs. In such cases, it is advisable to segment these critical assets and allow authorized communication.

Secure Remote Access with ZTNA

Remote access to the industrial OT environment is a growing need. Older generation factories were air-gapped with no connectivity with the outside world. With industry 4.0 adoption, more OT machines and equipment are now connected to the network. Remote access to these OT systems is required for various use cases such as

1. **Remote Troubleshooting:** Typical manufacturing plants have hundreds of machineries and components from different vendors.

These vendors might have their technical and support teams sitting in different countries. Bringing these technicians onsite to troubleshoot is extremely expensive and time-consuming. Halting production may result in huge losses to the company. More and more teams are looking to support remote troubleshooting in their OT environments. Having the ability to remotely diagnose and fix the issue helps restore the operations quickly without waiting for travel. However, such access needs to be highly controlled and given to specific machinery components for a fixed duration with the ability to log the actions.

2. **Regulatory Compliance and Predictive Alerts:** In some cases, regulatory requirements mandate constant monitoring of specified systems for compliance purposes. This is common in the energy sector where they need to comply with NERC CIP regulations. While this use case does not fall under specific individuals accessing the system, it still requires careful consideration in terms of network segmentation and allowing specific telemetry information to be shared outside the OT environment. A common way to implement such a use case is to have a dedicated telemetry network with one-way read-only communication toward the monitoring system. Some vendors also offer predictive alerts about their machinery by constantly monitoring the logs and performance of their machines. In such cases also, telemetry or IoT data is required to be shared outside the OT environment, usually to a cloud-based data analytics system.

In this chapter, we will focus on remote troubleshooting because that requires more controlled read-write access to OT machinery. Any unauthorized access or unaccountable change might bring the entire production network down. It is not uncommon to see the contractors or operations team install their own solutions, such as cellular gateways or broadband with direct access to the devices or via some form of VPN on the top. Such ad-hoc solutions pose security risks and are cumbersome to maintain. In some cases, it is more controlled where the IT provisions the VPN gateways up to the IDMZ level of the Purdue Model. Still, this approach has the following flaws:

- Provisioning VPN credentials on a need basis is an operational overhead.
- Providing access to a very specific machine is cumbersome.
- IDMZ firewall rules need to be configured and maintained.
- There is a lack of multifactor authentication.
- There is no visibility into the user activity.

Cisco solves this problem with Secure Equipment Access (SEA). This works by embedding the ZTNA gateway function directly into Cisco industrial switches and routers. Cisco SEA has a cloud-managed portal that centralizes the ZTNA gateway management and configuration of policies. This portal has wider functionality for IoT device management with SEA as one of the features. This cloud portal is known as the Cisco IoT Operations Dashboard (IoT-OD). Concerning remote access, the IOT-OD portal primarily acts as a ZTNA trust broker, which allows access to specific resources after authentication and authorization of the user.

Access to the OT devices can be provided in two modes:

- **Clientless ZTNA Mode:** In this case, the user needs to use a web browser to create remote sessions using RDP, VNC, HTTPS, and Telnet/SSH. These methods are allowed after the user is authenticated and authorized. This method does not allow the use of customer or native applications from the remote user desktop. This mode is useful when you need to access the device CLI or GUI.
- **Agent-Based ZTNA Mode:** This mode is also known as SEA Plus. This method creates a secure tunnel between the remote user computer and IoT-OD. The SEA agent/app needs to be installed on the remote user’s computer. A remote user’s computer has a virtual connection with the machine/device via the TUNTAP virtual network. The routes on the remote computer are then changed to use this TUN device. Any native application can be used to interact with the machine because the remote user has a direct tunneled connection with the device example; a native PLC programming application running on a user machine can be used to program the PLC.

Figure 21-11 shows the Secure Remote Access solution using Cisco Secure Equipment Service hosted on Cisco IoT-OD.

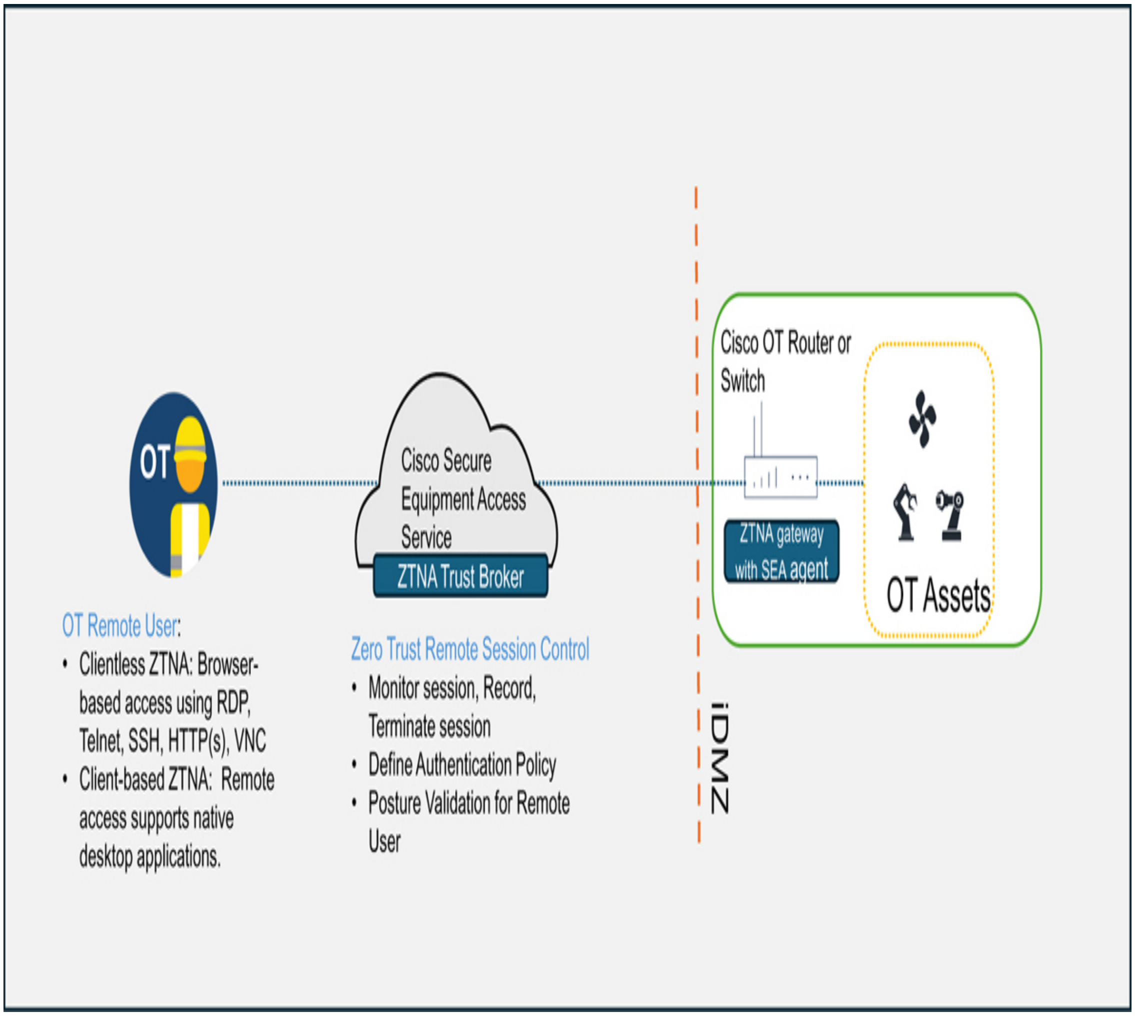

OT
Cisco Secure Equipment Access Service
ZTNA Trust Broker
Cisco OT Router or Switch
ZTNA gateway with SEA agent
OT Assets
iDMZ
OT Remote User:
• Clientless ZTNA: Browser-based access using RDP, Telnet, SSH, HTTP(s), VNC
• Client-based ZTNA: Remote access supports native desktop applications.
Zero Trust Remote Session Control
• Monitor session, Record, Terminate session
• Define Authentication Policy
• Posture Validation for Remote User

Figure 21-11 *Remote Device Access Using Cisco SEA Plus*

Note

Both SEA and SEA Plus can be used simultaneously. If the remote user does not have administrator access to the machine, it will make more sense to use SEA because no agent will have to be installed. Cisco IOT devices like routers or switches need to be onboarded to Cisco IoT-OD before remote access can be provided to the connected OT asset.

The steps to allow remote access to an OT asset via SEA/SEA Plus are as follows:

1. The network admin onboards the Cisco IoT router or switch to IoT-OD.

2. The network admin defines the assets that can be accessed via specific IoT switches—for example, a server with a specific IP address. The network admin also defines the access methods like SEA-SSH, SEA_Plus-All Protocols, SEA Plus—Modbus-TCP only.

3. The network admin then creates user/user groups with access methods that define access to specific assets in the OT environment. The admin primarily defines which remote user can access what and the duration of such access.

4. The remote user logs in to the IoT-OD SEA portal, selects the device to which access is required, clicks on Connect, and remote sessions are opened for the user by SEA.

5. The remote user has access based on the defined methods and policies defined on the IoT-OD Portal.

Another industry term, *remote privileged access management (RPAM)*, is primarily ZTNA with additional features and steps for security, such as scheduled access and the ability to view flows. IoT-OD, which acts as a zero trust broker for remote connections, offers the following controls to implement zero trust with RPAM.

1. **Scheduled Access:** The admin can define the access methods for groups of users that define the time, duration, and asset type that specific remote user groups can access.

2. **Device Posture:** Remote user security posture can be achieved by integrating Cisco Duo. This allows the admin to provide access to an OT asset only if the remote user machine complies with OT team security policies such as malware protection or an up-to-date operating system.

3. **Single Sign-On (SSO) and Multifactor Authentication (MFA):** Security standard bodies like NIST, ISA/IEC-62443, and NERC CIP recommend the use of MFA for remote access. Having the ability to attach MFA with remote access becomes even more important from

a compliance perspective. SSO and MFA can be integrated with the SEA for remote user login.

4. **Session Monitoring:** You can create different roles on SEA, such as the SEA admin and users. SEA admins can monitor who is connected, join any active connections, terminate any session, see session history, and view inline session recordings to audit what a remote user has done in a given remote session.
5. **Session Request and Approval:** A remote user can request the SEA admin to allow access to specific OT assets. For example, a remote user can ask for RDP access to a specific OT asset for a specific time via the IoT-OD portal. An authorized remote user can see the devices available via IoT-OD but cannot access it until the session request is approved.

Note

Session recordings are kept in the AWS S3 bucket, which is provided by the customer. These recordings are not stored on Cisco IoT-OD. This allows a customer to define the retention policies for these recordings based on their compliance and audit needs.

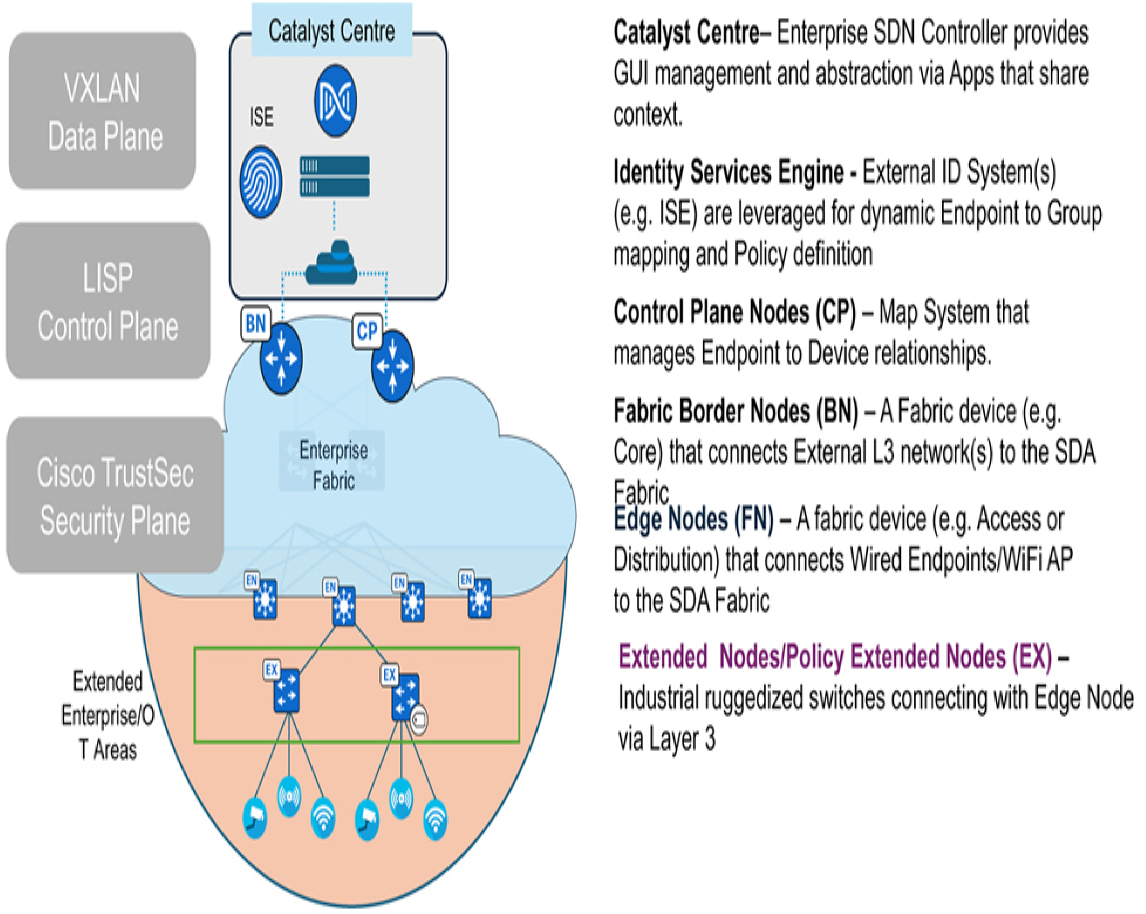
Extending ZTNA in a Noncarpeted Environment with Cisco SD-Access

This section covers how an enterprise can extend the zero trust concept beyond carpeted floors such as warehouses, and in OT-like areas such as HVAC and building management systems (BMS) that require the use of ruggedized industrial switches due to harsh environments.

Note

It is important to note that at the time of writing this chapter, Cisco does not support direct mapping between Purdue Model levels with the SD-Access solution. Cisco does not recommend using an SD-Access[nd]based solution for Purdue Levels 0 to 2. However, you can use SD-Access until Level 3 and also benefit from the assurance features for Cisco switches with Cisco Catalyst Center.

However, Cisco SD-Access is completely supported for scenarios where enterprise networks need to be extended beyond carpeted floors or requirements do not need Purdue Level 0[nd]2 mapping. Cisco SD-Access is a next-generation networking solution that leverages software-defined networking principles to automate and simplify network management. It provides comprehensive end-to-end segmentation, enhances security through consistent policy enforcement, and ensures seamless connectivity across wired and wireless environments. In contrast, Cisco SD-Access was primarily focused on enterprise and carpeted floor service. It supports the expansion of the SDA concept beyond carpeted floors into harsh industrial and OT environments. This is done by supporting industrial switches to be part of Cisco SD-Access as extended nodes (ENs) and policy-extended nodes (PENs). Figure 21-12 shows the Cisco SD-Access solution components for extended enterprise deployment.



Catalyst Centre– Enterprise SDN Controller provides GUI management and abstraction via Apps that share context.

Identity Services Engine - External ID System(s) (e.g. ISE) are leveraged for dynamic Endpoint to Group mapping and Policy definition

Control Plane Nodes (CP) – Map System that manages Endpoint to Device relationships.

Fabric Border Nodes (BN) – A Fabric device (e.g. Core) that connects External L3 network(s) to the SDA Fabric

Edge Nodes (FN) – A fabric device (e.g. Access or Distribution) that connects Wired Endpoints/WiFi AP to the SDA Fabric

Extended Nodes/Policy Extended Nodes (EX) – Industrial ruggedized switches connecting with Edge Node via Layer 3

Figure 21-12 *Cisco SD-Access Extended Network Components*

From a ZTNA perspective, it is important to note that Cisco TrustSec forms the security plane for the SD-Access solution. It is defined in three phases:

- **Classification:** The security group tag is assigned dynamically based on the device authentication or statically via mapping with IP, VLAN, or port profile.
- **Propagation:** After the classification of device or user, the SGT is propagated from where the classification took place to where enforcement action is invoked. This process is called *propagation*. Cisco TrustSec has two methods of SGT propagation: inline tagging and SXP.

  With inline tagging, the SGT is embedded into the Ethernet frame, specifically the VXLAN header of the encapsulated frame in Cisco SD-Access, and this requires specific hardware support. If a device does not support inline tagging, it can use SGT Exchange Protocol (SXP), which allows transport of SGT-IP mapping.
- **Enforcement:** This refers to the action taken to either allow or deny the traffic. The enforcement device takes the source SGT and looks it up against the destination SGT to check whether or not the communication is permitted.

Next, let’s explore the concepts of EN and PEN in detail because these allow the extension of SDA into the OT environment.

Extended Node (EN)

In extended node, specific industrial switches connect with the Cisco SD-Access Fabric Edge in Layer 2 mode. This connection is 802.1Q trunk EtherChannel. These extended nodes can be onboarded to the SDA fabric via plug-and-play. Endpoint authentication using 802.1x and MAB is supported on EN; this allows dynamic VLAN assignment to the switch port. Because SGT is the policy plane for Cisco SDA, all policy applications will happen only at the Fabric Edge. This means that any east-west traffic with source and destination on the same EN is not subjected to SGT-based policy enforcement. For example, you cannot block the communication between two PLCs in the same VLAN with SGT when connected to the same extended node because policy enforcement happens on the Fabric Edge.

Let’s examine an example of a policy application with extended nodes, as shown in Figure 21-13. In this example, Host 1 is connected directly to Fabric Edge FE1, which is trying to communicate with Host 2 connected to Fabric Edge FE2 via an extended node.

1. Host 1 is directly connected with FE1; in this case, the policy enforcement point is the Fabric Edge. When Host 1 completes the 802.1x authentication on the Fabric Edge, FE1 receives the dynamic VLAN-SGT mapping from the ISE.

2. Host 2 is connected via an extended node. When it completes the 802.1x authentication, ISE only returns the VLAN assignment because the extended node is not able to fetch and apply the SGT policies. Static IP-SGT mapping is created for the devices connected on the extended node. (You can also use VLAN-SGT mapping in case multiple devices are connected in same VLAN on extended node.)

3. The traffic from Host 1 is sent to FE 1 because the default gateway exists on the Fabric Edge node.

4. FE1 will consult the control plane on where to send traffic and ensure the traffic reaches the destination (the VXLAN endcap carrying SGT). In this case, it is sent to the other Fabric Edge node 2.

5. At FE2, the policy checks whether or not communication between SGT100 and SGT200 is allowed. If it is allowed, traffic is sent toward the extended node to be shared with the Host 2

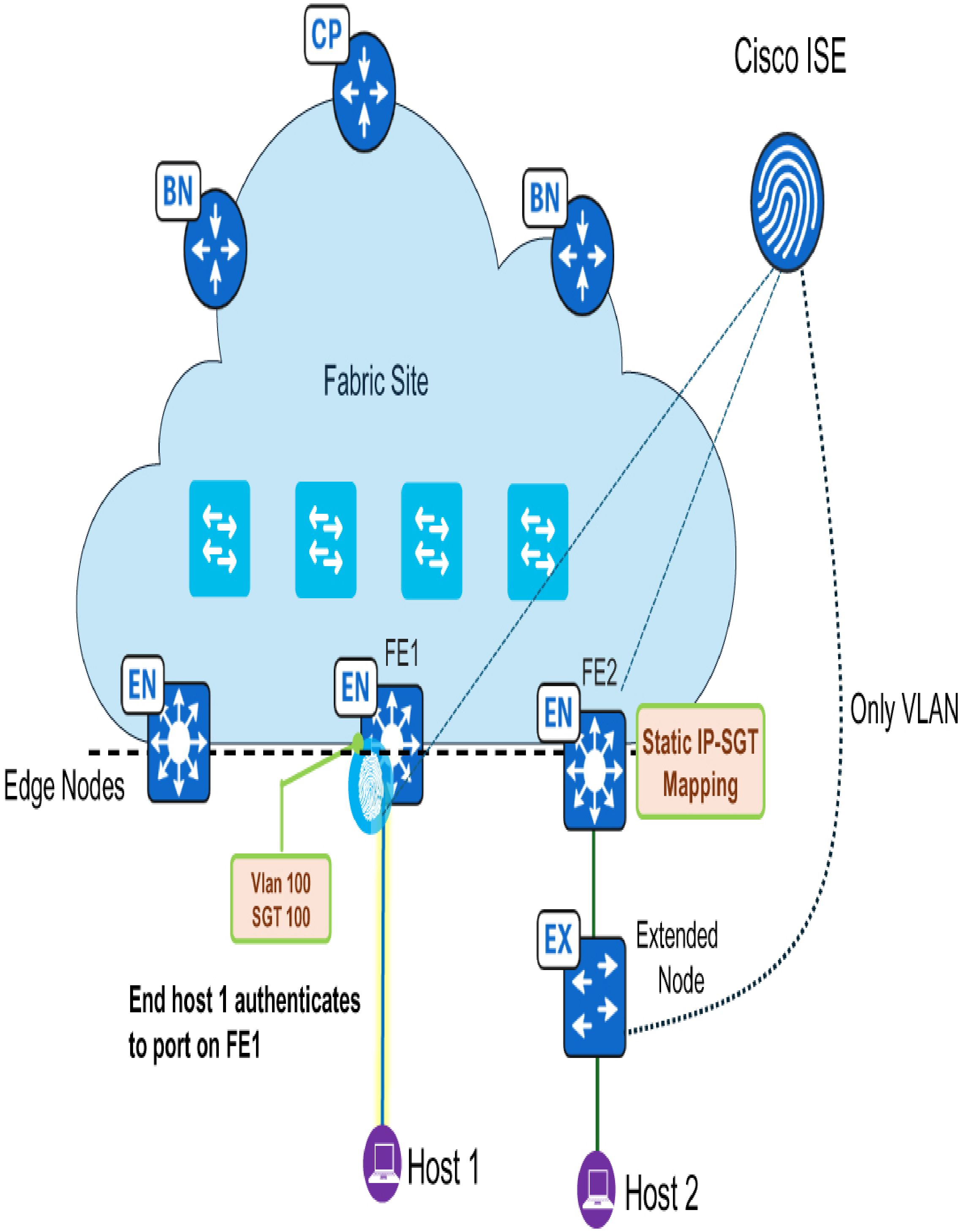


Figure 21-13 *Policy Application When Using Extended Nodes*

Policy Extended Node (PEN)

In policy extended node, selected industrial switch models can apply the SGT policy at the local port. In this case, the policy plane is extended to PENs. SGT policy contracts are directly downloaded on PENs, and SGACL policies for east-west traffic are applied locally on the PENs. For traffic that is destined outside the PENs, SGT is shared with the Fabric Edge using inline tagging. This mode is more suitable for use cases where microsegmentation is required at the local switch level.

Let's examine another example, as shown in Figure 21-14, where OT hosts connected to ENs and PENs are trying to communicate with each other. In this case, Host 1 is connected to the Fabric Edge via the policy extended node while Host 2 is connected via the normal extended node.

1. Host 1 authenticates using 802.1x with the PEN. In this case, Cisco ISE sends SGT+VLAN assignment directly to the PEN. The PEN downloads the SGACL and applies it locally.
2. Host 2 authenticates with the ISE; only dynamic VLAN is assigned to Host 2.
3. PEN shares the Host 1 SGT details with Fabric Edge 1 via inline tagging.
4. FE1 identifies the destination FE2 from the control plane as a destination for the packet, encapsulates in VxLAN, and sends it to FE2.
5. At FE2, the frame is removed, and SGACL is applied based on the ISE policy definition.
6. If allowed, the frame is then sent to the extended node and then to Host 2.

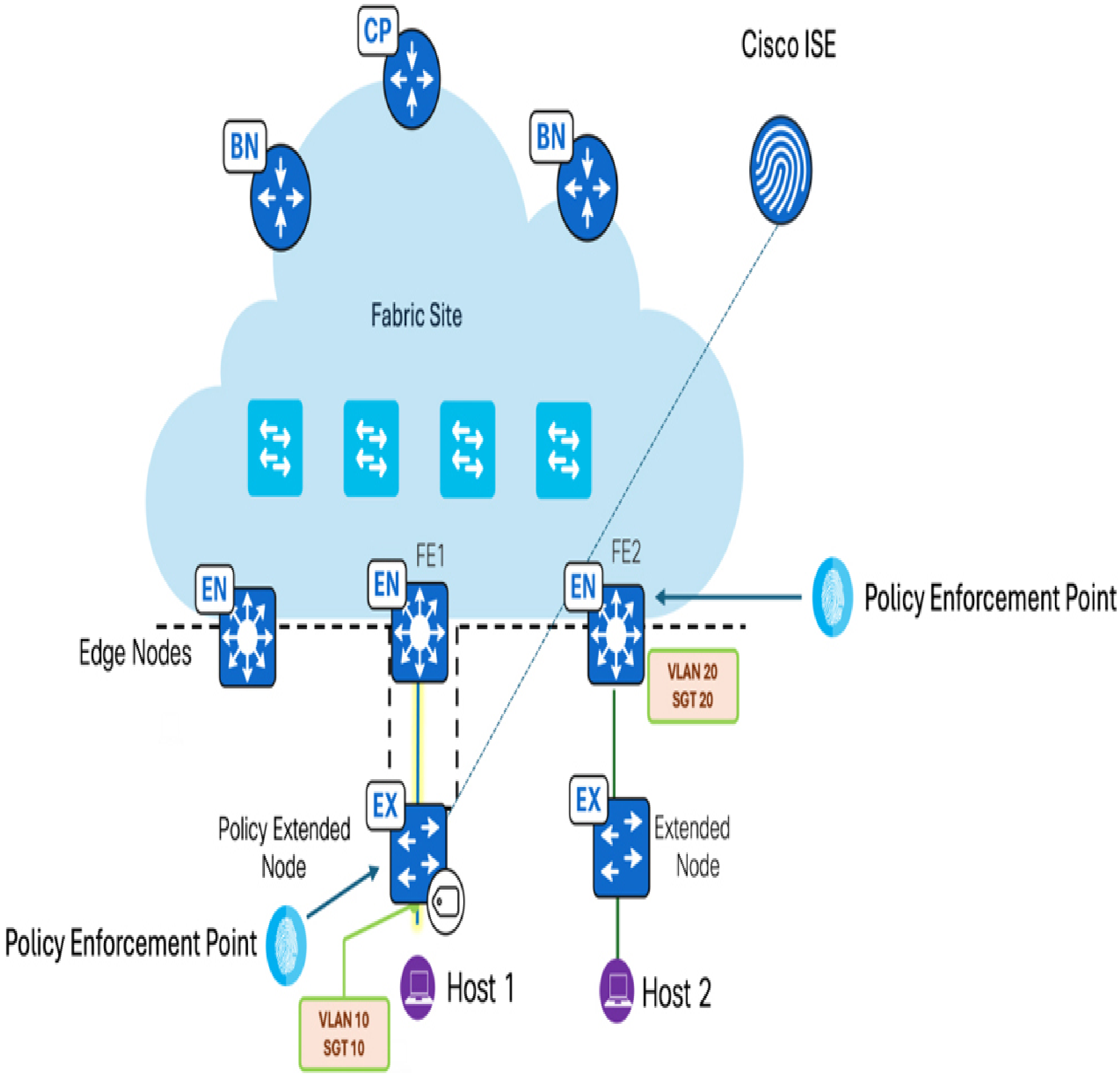


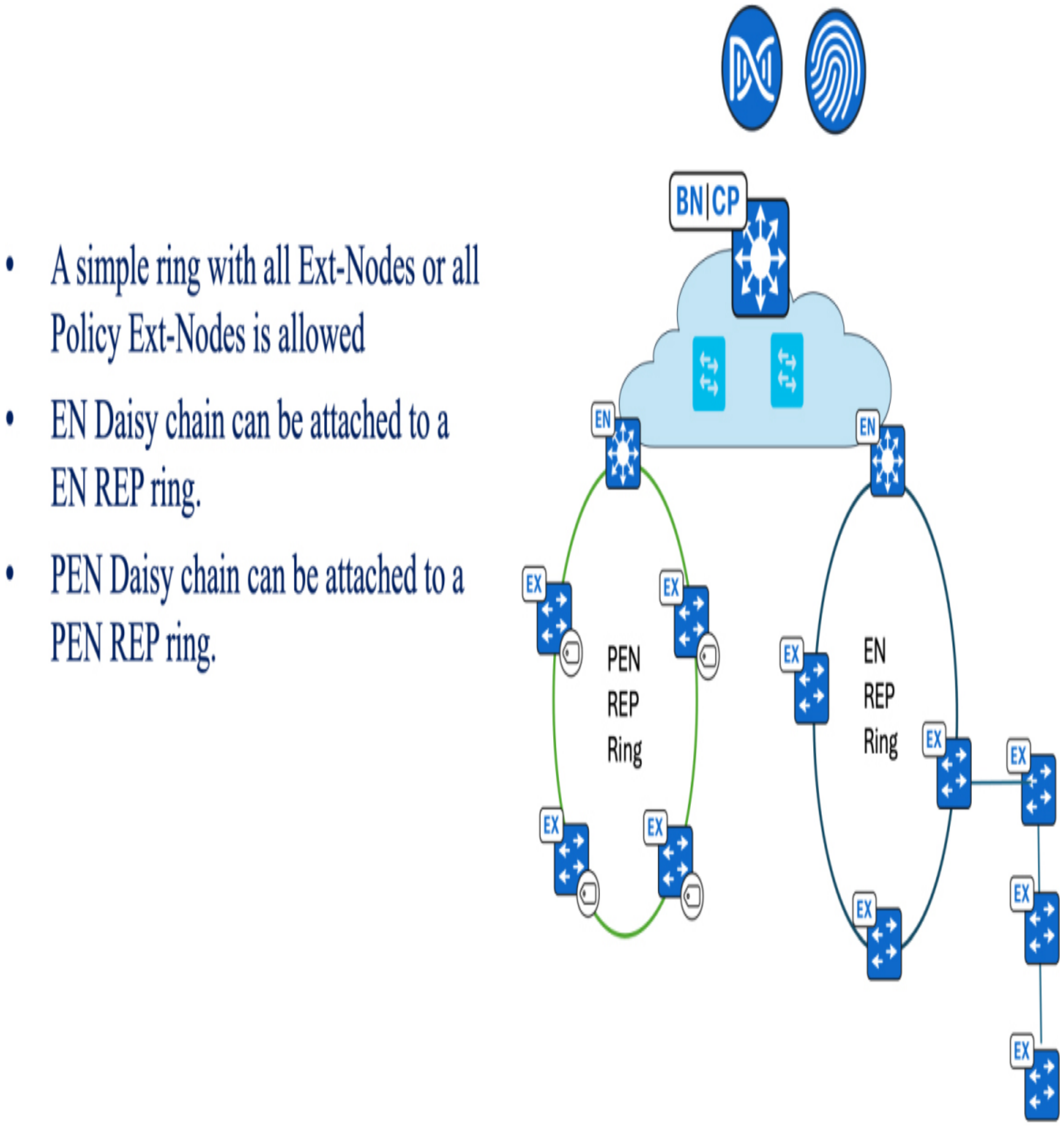
Figure 21-14 *Policy Application Using Policy Extended Nodes*

If Host 2 was connected at the same PEN as Host 1, SGACL would have been applied locally, and communication would be blocked or allowed based on the ISE policy.

Note

All the latest models of Cisco industrial switches are capable of PEN mode, and it is the default mode when booted. You can still move them to EN mode, but legacy hardware cannot be updated from EN to PEN mode. For PEN mode support on industrial switches and license requirements, please refer to the latest Cisco documentation.

With PENs, you can easily extend the segmentation directly into the harsh OT environment. These plant networks are usually spread in larger areas, and you will require a large number of switches to provide connectivity. Because high availability is critical to industrial use cases, ring topology is a common deployment model for OT switches. These rings could be further extended with daisy-chained industrial switches. Industrial OT uses specific protocols like Media Redundancy Protocol (MRP) and Resilient Ethernet Protocol (REP) for ring topology. Figure 21-15 shows how REP rings can be created using extended and policy extended nodes. These rings could also be extended using a daisy chain as per design requirements. These protocols have much faster convergence times compared to standard Spanning Tree[nd]based rings. Cisco Catalyst Center has workflows that automate the REP ring onboarding. It also supports dynamic addition or deletion of the nodes in the REP ring. Details of these protocols are beyond the scope of this book, but you need to understand how the zero trust constructs of segmentation will work when a ring of PENs and ENs is used.



- A simple ring with all Ext-Nodes or all Policy Ext-Nodes is allowed
- EN Daisy chain can be attached to a EN REP ring.
- PEN Daisy chain can be attached to a PEN REP ring.

Figure 21-15 *REP Rings Using Extended and Policy Extended Nodes*

When you put the PENs or ENs in a ring, ensure you do not mix them in a single ring. Mixing PENs and ENs will create an inconsistent policy plane because only PENs can apply the policies locally while ENs have to depend on the Fabric Edge. You can extend the rings with daisy-chained PENs.

For specific OT use cases, multiple design approaches can be adopted with Cisco SD-Access. A detailed discussion of this topic is beyond the scope of this book. Figure 21-16 shows a typical approach when adopting Cisco SD-Access solution for OT environments.

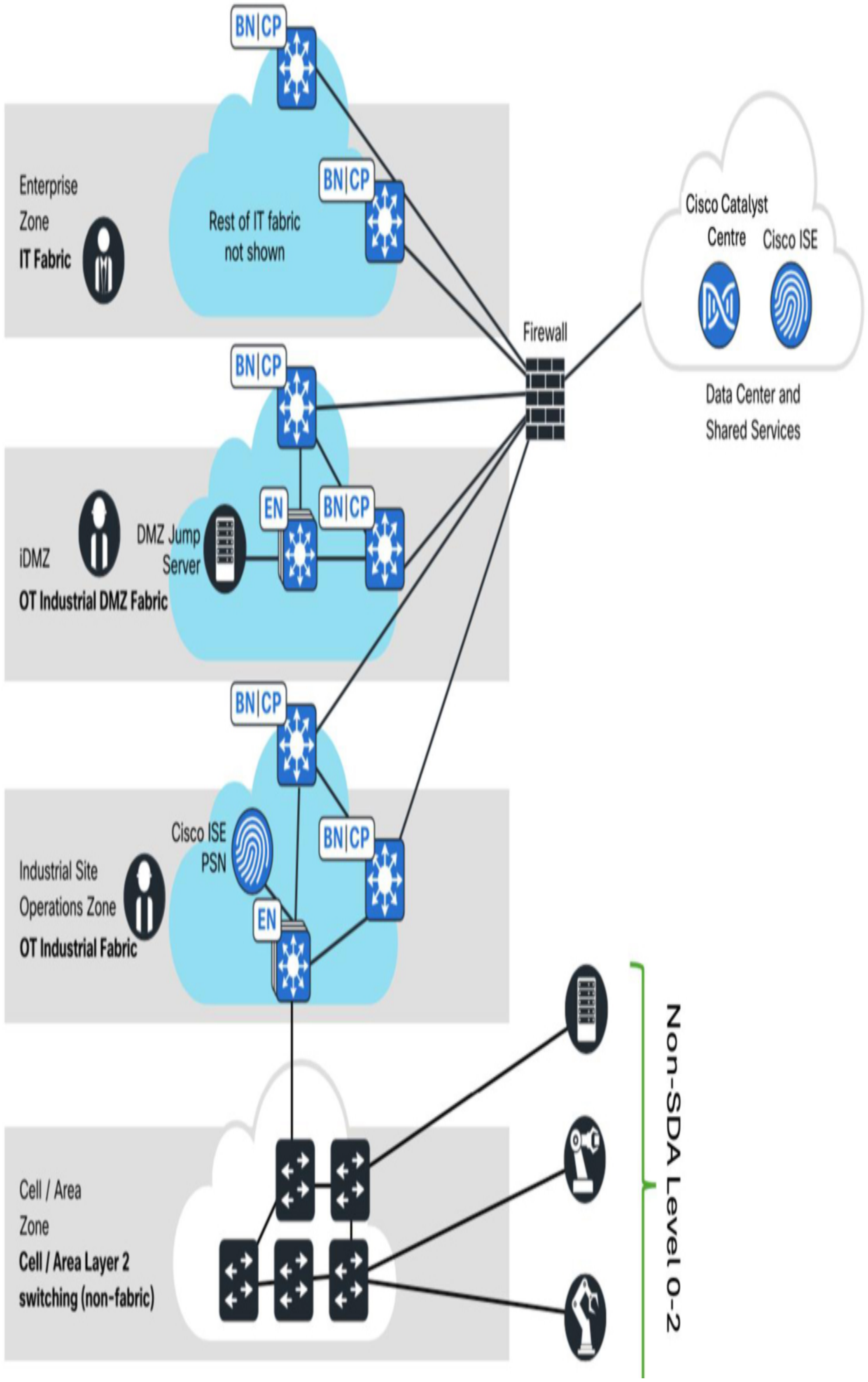

BN|CP
BN|CP
Enterprise Zone
IT Fabric
Rest of IT fabric not shown
Cisco Catalyst Centre
Cisco ISE
Firewall
Data Center and Shared Services
BN|CP
EN
BN|CP
DMZ Jump Server
iDMZ
OT Industrial DMZ Fabric
BN|CP
Cisco ISE PSN
BN|CP
EN
Industrial Site Operations Zone
OT Industrial Fabric
Cell / Area Zone
Cell / Area Layer 2 switching (non-fabric)
Non-SDA Level 0-2

Figure 21-16 *Cisco SD-Access Fabric Mapping with Purdue Zones*

Summary

Industrial manufacturing plants follow the Purdue Enterprise Reference Architecture for segmenting and grouping assets with similar needs. Macrosegmentation is achieved by using industrial DMZ between IT and OT areas. No direct access is allowed between them. To achieve microsegmentation in an OT network, you need to have deep visibility into device communication patterns, OT protocols, and OT-specific attributes. Cisco Cyber Vision provides these attributes, and microsegmentation can be done using Cisco ISE. Secure Equipment Access is one of the key requirements for OT assets. Cisco SEA and SEA Plus allow ZTNA-based remote access to the OT assets. Lastly, you can use Cisco industrial switches to extend or create separate OT SD-Access domains for your harsh environment that requires ruggedized devices with specific OT protocol support.

Part 6: Integrations and Automation

Chapter 22. Third-Party SDN Integrations

In this chapter, you will learn about the following:

- End-to-end policy strategy
- Multivendor campus environments
- Multivendor BGP EVPN and policy propagation
- Firewall connectivity in the campus
- Third-party vendor firewall policy integration
- Highly resilient third-party vendor firewall integrations

Introduction to Third-Party SDN Integrations

As organizations increasingly adopt zero trust architectures (ZTAs), ensuring consistent security across diverse multivendor networks has become crucial. This chapter will delve into achieving effective end-to-end policy propagation in environments where varied technologies and protocols must interoperate seamlessly. While zero trust demands uniform policy enforcement, the integration of different vendors introduces significant complexity. Here, we will address these challenges and provide strategies for maintaining consistent security across all network components.

We will also explore the pivotal role of security components, which, while essential for data protection, can pose integration challenges in heterogeneous setups. Through practical examples, this chapter will offer insights and strategies for building a flexible, resilient zero trust enterprise network that meets the demands of modern networking environments.

End-to-End Policy Strategy in a Multivendor Environment

In today's enterprise networking landscape, securing end-to-end segmentation across diverse environments—such as campus networks, SD-WAN, and data centers—is vital for robust security. Microsegmentation, particularly through the use of security tags, is a concept championed by Cisco and increasingly adopted by other vendors. This approach allows for the classification and enforcement of policies across various network segments, ensuring security requirements are consistently met wherever data flows within the network.

A primary method for achieving this goal is through Cisco's TrustSec framework, which utilizes security group tag (SGT) propagation. However, implementing end-to-end segmentation with SGT propagation presents challenges, especially when integrating Cisco devices with equipment from other vendors, each adhering to different standards and protocols.

In the following sections, we'll explore technical strategies for implementing end-to-end policy propagation in a multivendor environment, ensuring that SGTs are maintained across various domains. We'll also examine the complexities of securing heterogeneous environments, offering insights and practical examples to guide you through the process.

Benefits of End-to-End Segmentation

End-to-end segmentation enhances security by ensuring SGTs are consistently applied across all network domains. This approach limits access to designated resources to only authorized users or devices, reducing exposure to sensitive data and minimizing the risk of unauthorized lateral movement within the network.

Additionally, this segmentation strategy improves organizational agility and adaptability. New applications and workloads can be deployed rapidly without extensive reconfiguration (e.g., ACL firewall rules, provisioning new VLANs) across multiple devices, simplifying network operations and accelerating the time-to-service even in complex, segmented environments.

Microsegmentation advances this capability further by providing more granular traffic control compared to macrosegmentation. By applying security policies at the workload level, microsegmentation offers tighter control and reduces attack surfaces. Unlike macrosegmentation, which often involves complex provisioning like VRFs, microsegmentation allows for a more agile and less resource-intensive solution.

A centralized, identity-based management system (such as Cisco Identity Services Engine, or ISE) can streamline this process by serving as a single source of truth for classification across multiple domains. Centralized management reduces policy deployment complexity by automatically propagating changes throughout the network, making it easier to enforce consistent policies.

Finally, data plane tagging optimizes resource utilization by leveraging a device’s ternary content-addressable memory (TCAM) resources, ensuring tags are maintained as traffic moves through the network, as shown in Figure 22-1. This method eliminates the need for control plane protocols to reapply tags, offering a scalable and efficient solution that is ideal for large networks where minimizing resource usage is critical.

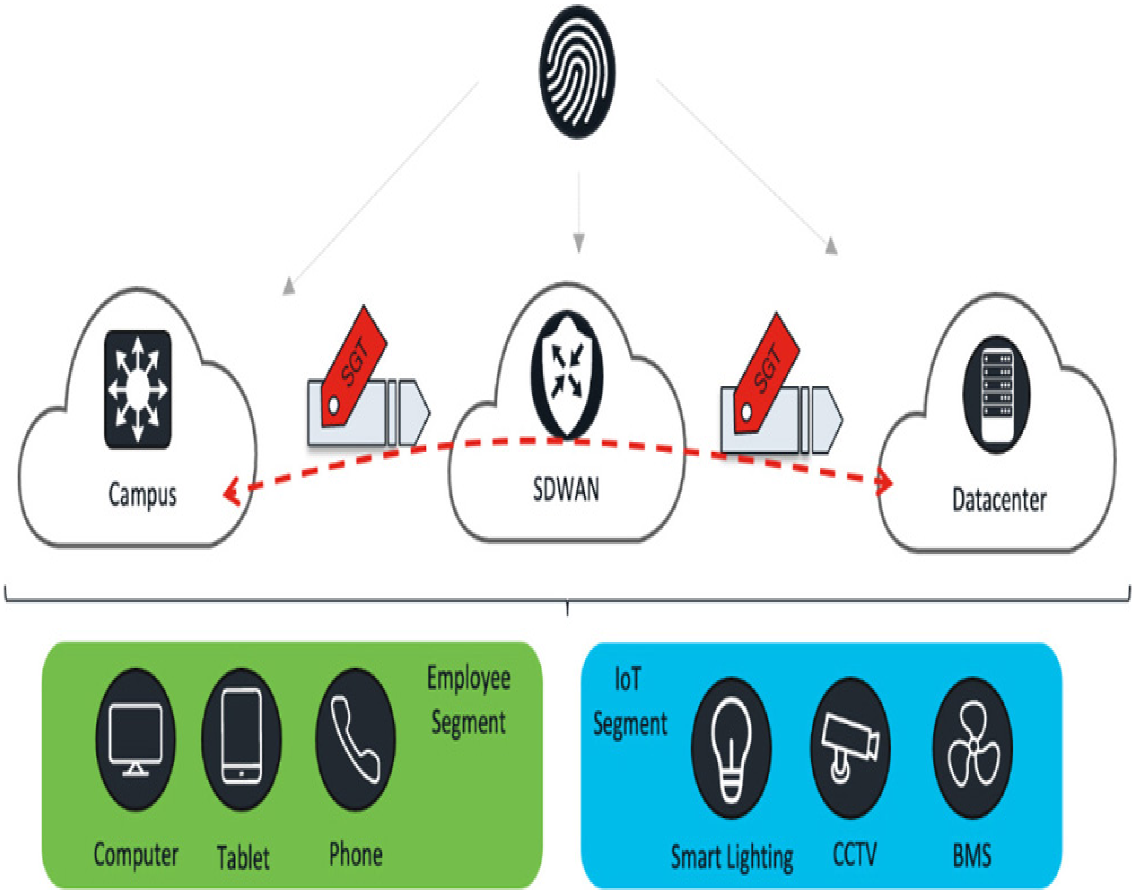


Figure 22-1 *End-to-End Policy Propagation*

Challenges in Multivendor Environments

Typical challenges in multivendor environments include the following:

- **Interoperability Concerns Across Vendors:** When you're working with a mix of devices from different vendors, achieving seamless end-to-end segmentation can be quite the puzzle. Each piece of equipment might handle security group tags in its own unique way. This means you often have to dive into extra configuration and testing to make sure these tags get propagated and enforced consistently, no matter which manufacturer's gear is in play.
- **Resource Constraints and TCAM Limitations:** Data plane tagging is a slick way to manage things, but it does have its limits. In larger networks with tons of segmentation policies, TCAM can quickly become a pinch point. It's crucial to juggle these resources wisely to

keep SGT propagation scalable without bogging down network performance.

- **Operational Complexity in Policy Propagation:** Managing identity-based policies from a central location is great in theory, but putting it into practice—especially across multiple vendors—can get tricky. It requires strong integration with what you already have to ensure that classification is spot-on and policies are applied consistently.
- **Scalability Challenges with Control Plane Tag Propagation:** Control plane protocols like the Security Group Tag Exchange Protocol (SXP) can help with SGT propagation, but they come with their own set of challenges. They tend to add layers of complexity and can struggle to keep up in large, constantly changing environments.
- **Inline Tag Propagation:** This process involves the transmission of security group tags within the data plane, such as within Cisco Metadata (CMD) or VXLAN headers. Cisco employs this method for SGT tagging, but other vendors might use proprietary techniques for tagging, interpreting, or propagating SGTs.

Consider an organization aiming to implement a zero trust strategy across a multivendor network, ensuring consistent end-to-end policy propagation across all devices, regardless of vendor. A crucial part of this strategy involves maintaining SGT propagation throughout all network segments, covering branch offices, data centers, and inter-region connections. Due to the diverse environment, this includes compatibility with both Cisco and non-Cisco devices used across the campus.

The organization is gradually migrating some locations to SD-Access (Type Site A), while others continue to operate on traditional campus networks (Type Site B). Additionally, certain offices that rely on non-Cisco devices (Type Site C) cannot transition to SD-Access.

To achieve secure WAN connectivity across branches, the organization has selected Cisco SD-WAN as its primary solution. This deployment supports SGT propagation within the SD-WAN environment. However, for inter-regional connectivity, the organization depends on a third-party service provider that does not support SGT propagation over its backbone. Despite this, the organization is committed to maintaining a consistent zero trust strategy across regions, exploring alternative methods for policy propagation where SGT tagging is not supported. The goal is to enforce a unified zero trust policy across all locations, as shown in Figure 22-2, ensuring robust security even in sites that will not migrate to SD-Access.

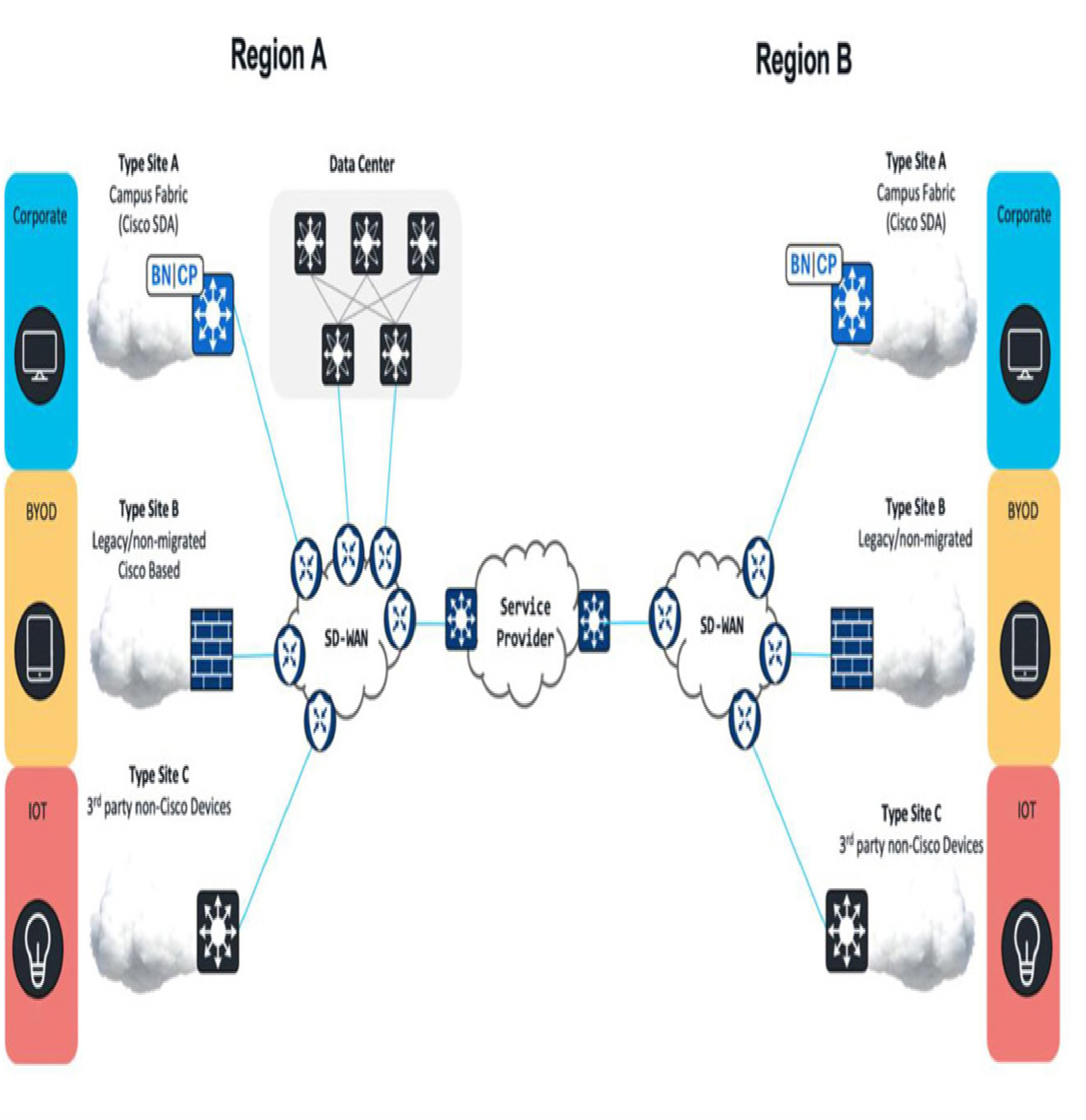


Figure 22-2 *Organization Networking Requirements*

Let’s explore how to ensure comprehensive policy propagation in a mixed environment where multiple networking technologies coexist alongside a combination of Cisco and non-Cisco devices, all requiring seamless integration.

Scenario 1: Site Type A with Cisco SD-WAN Integration

Site Type A represents a branch office that has already migrated to SD-Access technology, which inherently supports inline tagging security group tag propagation. Endpoint traffic classification and tagging are achieved directly at the Fabric Edge level, with SGTs embedded into VXLAN encapsulation within the SD-Access solution's data plane. Architecturally, north-south traffic, directed to other sites or to data centers/the Internet, is handled by Border nodes configured with Layer 3 (L3) handoffs. Each virtual network can be mapped into different VPN IDs on the SD-WAN side. The translation of SGTs from the VXLAN header occurs using the CMD header, which retains the SGT and maps it into the SD-WAN IPsec tunnel header, maintaining SGT continuity over SD-WAN transmission; this carriage of SGT information is further shown in Figure 22-3.

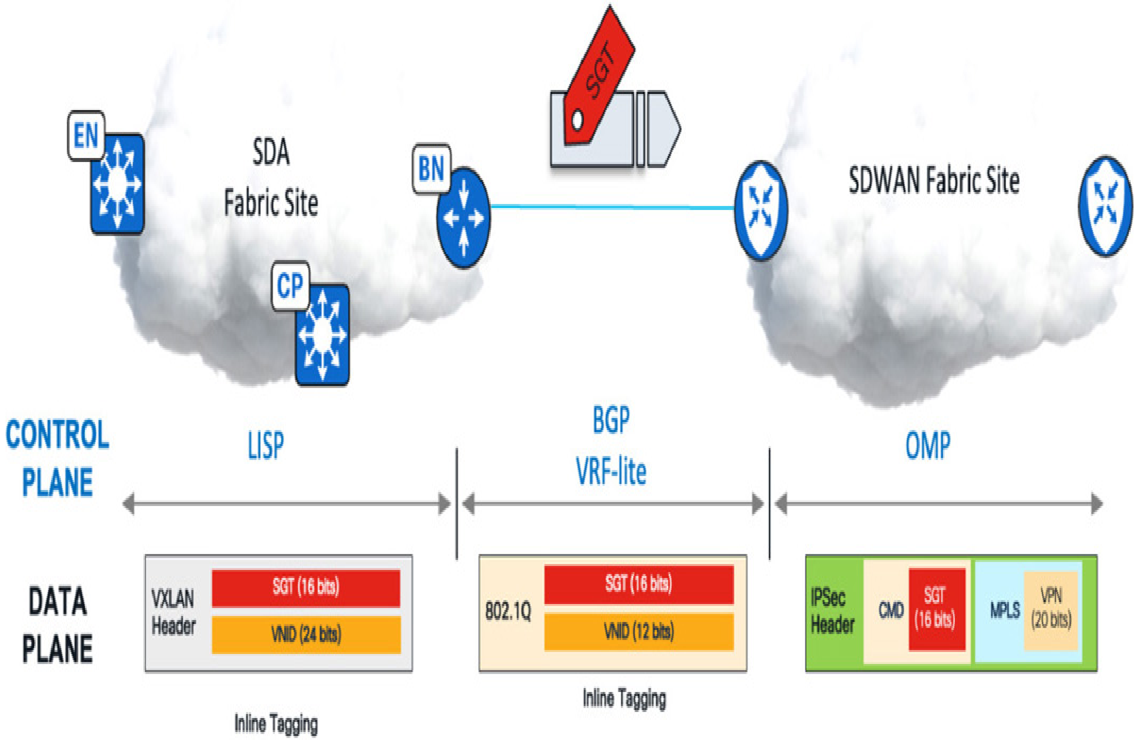


Figure 22-3 *Site Type A with Cisco SD-WAN Policy Propagation*

Scenario 2: Site Type B with SD-WAN Integration

In a traditional Cisco campus network, keeping security group tag propagation going smoothly is usually a breeze. The reason is that most modern Cisco switches and routers support CMD header propagation, ensuring that policies are enforced correctly across the network. But things can get tricky when you mix in devices from other vendors—like security firewalls, which are pretty common—or even older Cisco platforms that lack CMD support.

Cisco's TrustSec framework steps in to tackle these challenges, mainly using the SGT Exchange Protocol over TCP (SXP) protocol and SGT caching mechanisms. When you have a device that can't handle CMD headers, SXP comes to the rescue. It lets you exchange the necessary policies with the identity server, allowing SGTs to be mapped to IP addresses (and the other way around) on devices linked to those non-CMD-capable pieces of equipment.

Another handy tool is SGT caching, which enables direct peer-to-peer connections between edge devices to share SGT-IP mappings. This method helps ease scalability issues, particularly when it comes to the number of SXP sessions that the identity management system can handle at once. Plus, it cuts down on TCAM resource use, since mappings are created based on actual traffic rather than preloading the whole IP-SGT map.

We'll dive deeper into integrating firewalls in these setups in the following sections. For SD-WAN edge devices, exchanging SGT information is typically managed similarly to SD-Access border devices, by mapping SGTs shown in Figure 22-4 into the IPsec header.

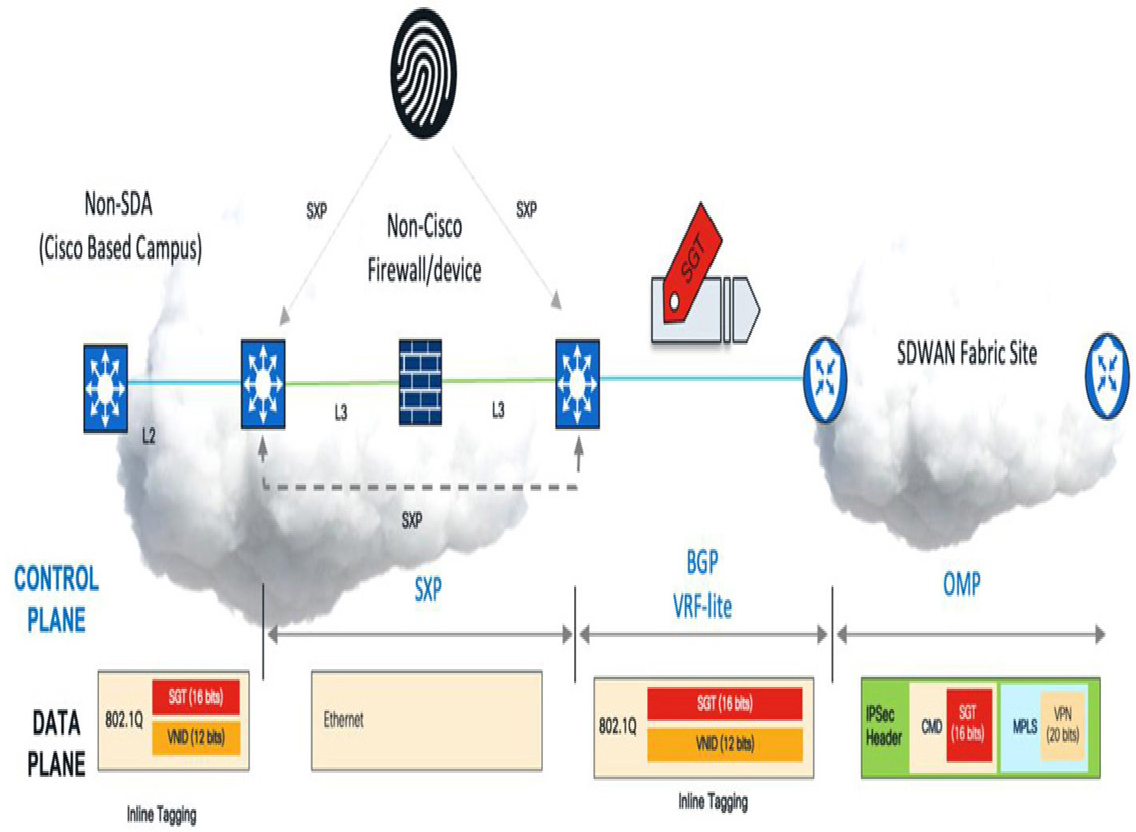


Figure 22-4 *Site Type B with Cisco SD-WAN Policy Propagation*

Scenario 3: Site Type C with SD-WAN Integration

For campuses that rely on third-party equipment, VXLAN-EVPN is becoming a go-to solution for building multivendor networks. This technology offers a standardized framework for overlay services, making it easier for different devices to work together seamlessly. One of the standout features of VXLAN-EVPN is its capability to pass SGT information in the data plane, ensuring consistent segmentation and policy enforcement across the network, no matter which vendor's equipment you're using.

Built on open standards, VXLAN-EVPN offers flexibility and supports scalable, vendor-agnostic deployments. Its overlay capabilities simplify the complexities of Layer 2 and Layer 3 connections, making it an ideal choice for robust, scalable solutions in large, distributed campus environments. With VXLAN-EVPN, as shown in Figure 22-5, campuses can enjoy a more streamlined and efficient network setup, tailored to handle diverse equipment while maintaining strong security and performance.

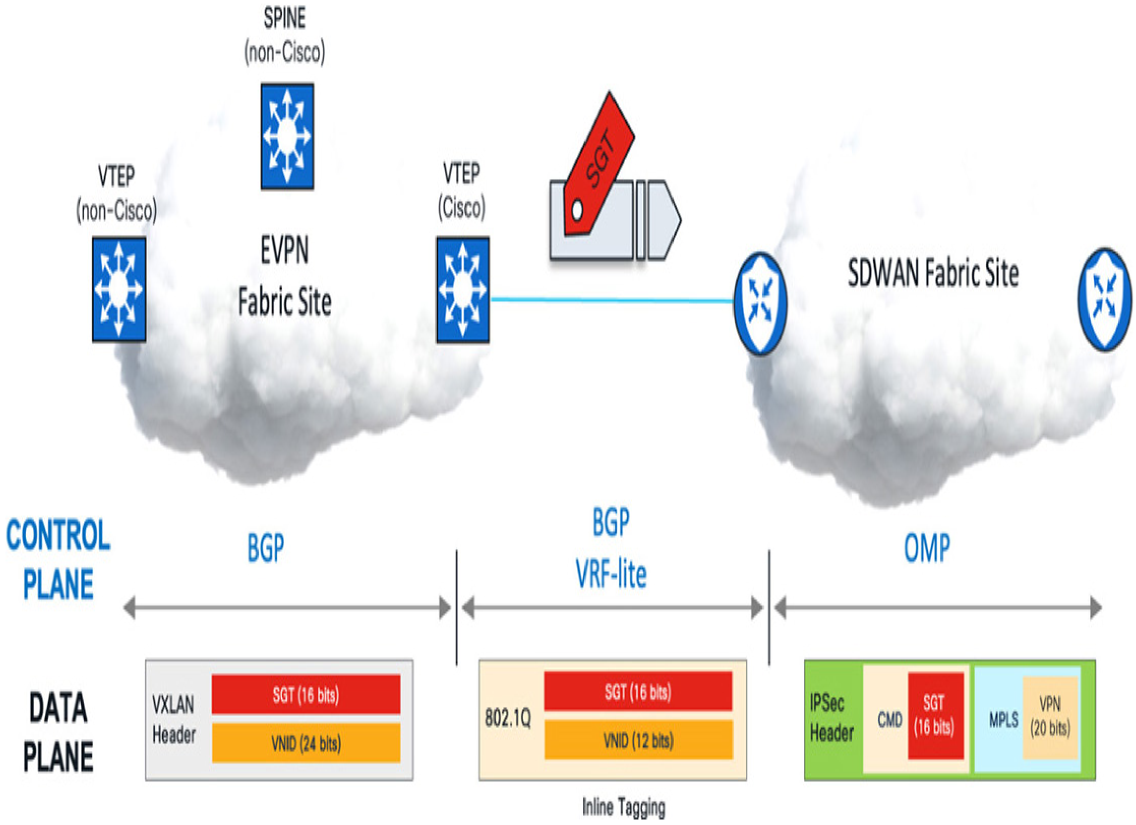


Figure 22-5 *Site Type C with Cisco SD-WAN Policy Propagation*

Scenario 4: Inter-Regional Communication

When inter-regional connectivity is required, external service providers are commonly used. These providers typically offer an IP transport network that lacks support for inline tagging and is limited to the IP layer, without additional Ethernet header extensions. Because this connectivity spans multiple regions, relying on traditional SGT-to-IP mapping can quickly become inefficient and unscalable.

Ethernet VPN (EVPN), with its scalability benefits through VXLAN encapsulation, is an ideal solution in this context. As shown in Figure 22-6, it enables seamless SGT propagation and consistent, end-to-end policy enforcement without the need for extra mapping or isolated segments.

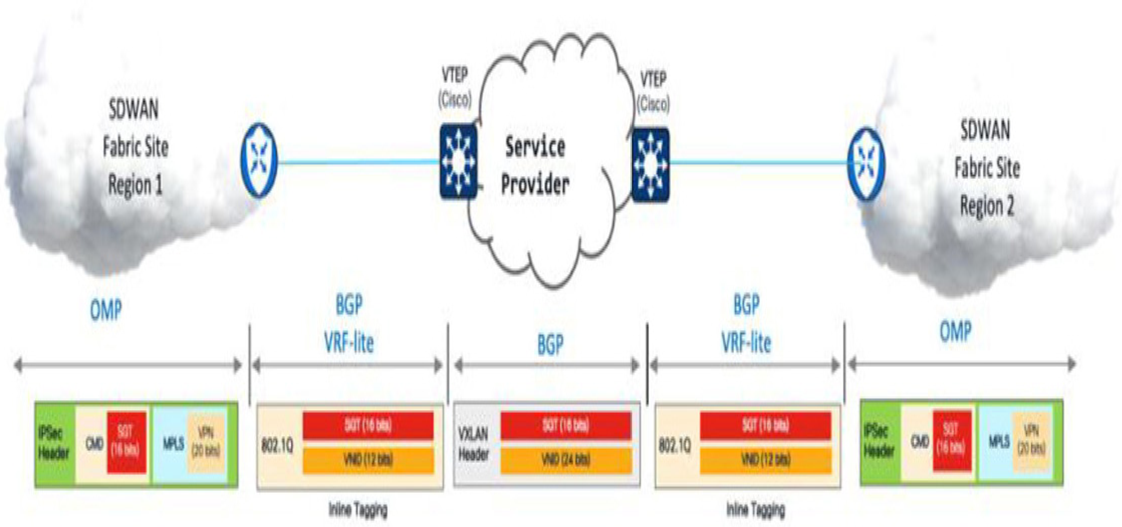


Figure 22-6 *Inter Region Policy Propagation*

Why VXLAN-EVPN?

As organizations seek scalable ways to propagate security group tags, transporting SGTs in the data plane often emerges as the most efficient solution. In Cisco networks, this is typically achieved through CMD headers for campus networks or VXLAN overlay headers in SD-Access deployments. However, since these methods are Cisco-specific, they can present challenges when integrating with non-Cisco devices.

In multivendor environments, SXP is frequently used at network edges where CMD headers aren't supported. While SXP can facilitate SGT propagation under these conditions, it can quickly become a bottleneck as the network expands and the number of segmentation policies increases. SXP also introduces additional operational complexity because it requires managing numerous peer connections and maintaining mapping relationships. This can lead to further scalability challenges, such as limits on the number of SXP sessions that can be supported or the need to download extensive mappings for all IP-to-SGT assignments—consuming TCAM resources—even if some of these mappings are rarely used.

To overcome these limitations and provide a scalable, multivendor-friendly solution, VXLAN-EVPN emerges as an ideal approach. This open-standard protocol leverages VXLAN encapsulation with the VXLAN Group Policy

Option (GPO) shown in detail in Figure 22-7, enabling seamless SGT tagging and policy propagation across heterogeneous networks. Here’s why VXLAN-EVPN is technically well-suited for this task:

- **Multivendor Support:** When it comes to Cisco support, VXLAN-EVPN initially originated in data center environments and was first implemented on platforms within Cisco’s data center portfolio. However, it has since evolved beyond data centers, extending into areas like campus networks. Today, VXLAN-EVPN has become a widely supported standard across various domains, with an increasing number of vendors incorporating EVPN capabilities into their products. This widespread adoption facilitates seamless interoperability across diverse networking platforms, making VXLAN-EVPN an ideal choice for multivendor environments.

- **BGP-Based Policy Propagation for Scalability:** VXLAN-EVPN leverages Multiprotocol BGP (MP-BGP) as its control plane to distribute network reachability information. This means it can handle both Layer 2 (MAC) and Layer 3 (IP) address information, making VXLAN-EVPN a versatile solution for various networking needs, whether in large-scale campus networks, SD-WAN setups, or data centers.

- **VXLAN Encapsulation for Overlay Simplicity:** Like Cisco’s SD-Access, VXLAN-EVPN uses VXLAN encapsulation to transport traffic across the network. VXLAN provides a Layer 2 overlay over an IP-based underlay, which simplifies the network architecture by decoupling the physical infrastructure from the logical segments. The encapsulation itself is open standard, which ensures interoperability across different vendors, allowing any device that supports VXLAN to participate in the overlay network.

- **Standardized VXLAN Group Policy Option (GPO):** The VXLAN Group Policy Option (VXLAN-GPO) brings a standardized way to embed security group tag information directly into the VXLAN header. According to IETF drafts, this method adds a header field to the VXLAN frame that carries important metadata, like security group information, without needing control plane support from intermediate devices.

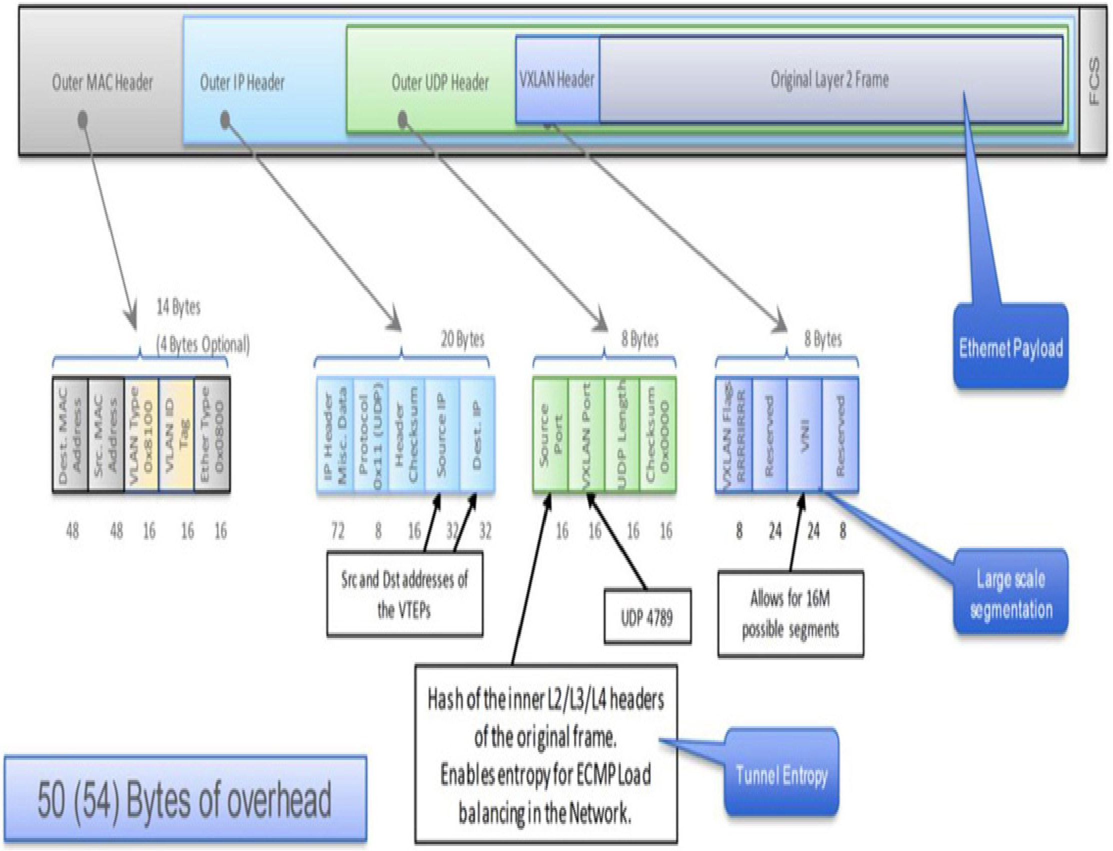


Figure 22-7 *VXLAN-GPO*

VXLAN-GPO is a major advancement for multivendor environments. It ensures segmentation policies are consistently enforced across the network, regardless of the hardware being used. With security tags traveling with the packet inside the VXLAN header, enforcement is achieved end-to-end, reducing the need for additional control plane mechanisms such as SXP.

Moreover, the use of 24-bit VXLAN network identifiers (VNIDs) allows for the creation of nearly 16 million unique segments. Such scalability makes VXLAN-EVPN particularly beneficial for enterprises that must support vast numbers of users, devices, and services across different domains.

Overall, VXLAN-GPO not only simplifies policy enforcement but also significantly boosts network scalability, making it an excellent choice for modern, dynamic networking environments.

BGP EVPN Detailed Traffic Flow and Architecture

VXLAN-EVPN operates within a spine-and-leaf architecture (typically) using virtual tunnel endpoints (VTEPs) and route reflectors to manage traffic flows efficiently. This approach, shown in Figure 22-8, supports both macrosegmentation and microsegmentation, which is essential for handling the diverse traffic flows in modern enterprise networks.

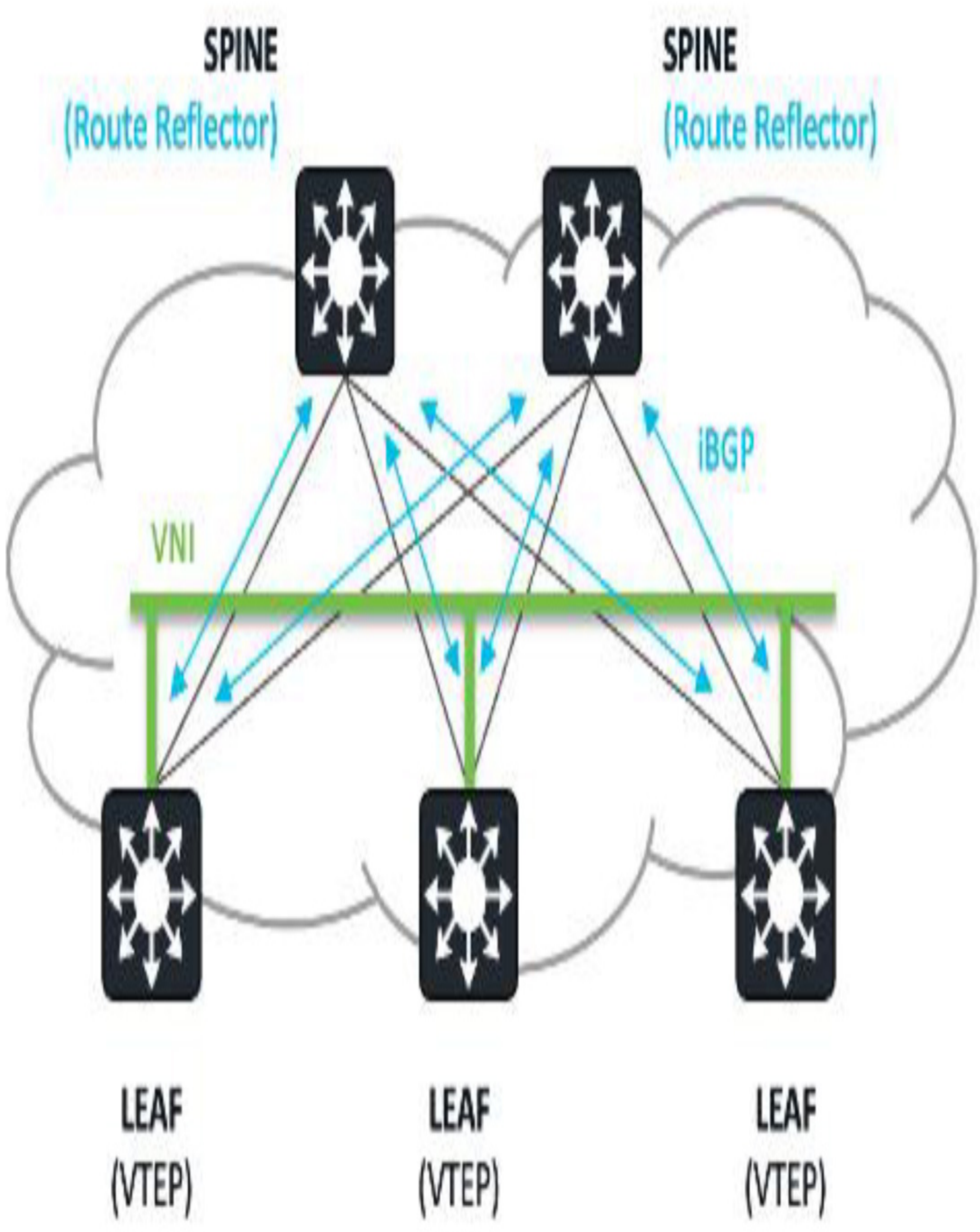


Figure 22-8 *VXLAN-EVPN Reference Architecture*

- **Virtual Tunnel Endpoints (VTEP):** VTEPs, typically located at the network edges, encapsulate Ethernet frames within VXLAN headers, enabling Layer 2 connectivity over Layer 3 networks. Each VTEP has a unique IP address that serves as the source or destination for VXLAN traffic, allowing SGTs to be preserved across network segments.
- **Virtual Network Identifier (VNI):** VNI is a unique identifier used to distinguish different virtual networks within the EVPN overlay. It maps traffic within a specific tenant's network segment, allowing multiple isolated Layer 2 and Layer 3 networks to coexist over a shared underlay infrastructure
- **Route Reflectors for Centralized Control:** By centralizing BGP route advertisement, route reflectors reduce the complexity of iBGP sessions within the EVPN fabric. This setup enhances scalability and minimizes the operational overhead associated with maintaining a full mesh of BGP sessions between VTEPs.
- **iBGP Sessions for Efficient Traffic Forwarding:** iBGP sessions are established between VTEPs and route reflectors, enabling the exchange of routing information necessary for segmenting and directing traffic based on SGTs. This ensures that each VTEP is aware of endpoint locations and policy details, allowing for efficient traffic forwarding across the network.

The EVPN architecture is all about flexibility when it comes to implementing overlay and VNI functionality. While most networks rely on the popular spine-and-leaf setup, there's an alternative that might just suit your needs: the leaf-to-leaf architecture.

In this leaf-to-leaf configuration, you establish the EVPN overlay using just two devices that act as VTEPs, skipping the need for route reflectors or spine nodes altogether. This streamlined approach, shown in Figure 22-9, is tailored to specific requirements, offering a simplified solution that's efficient for certain scenarios.

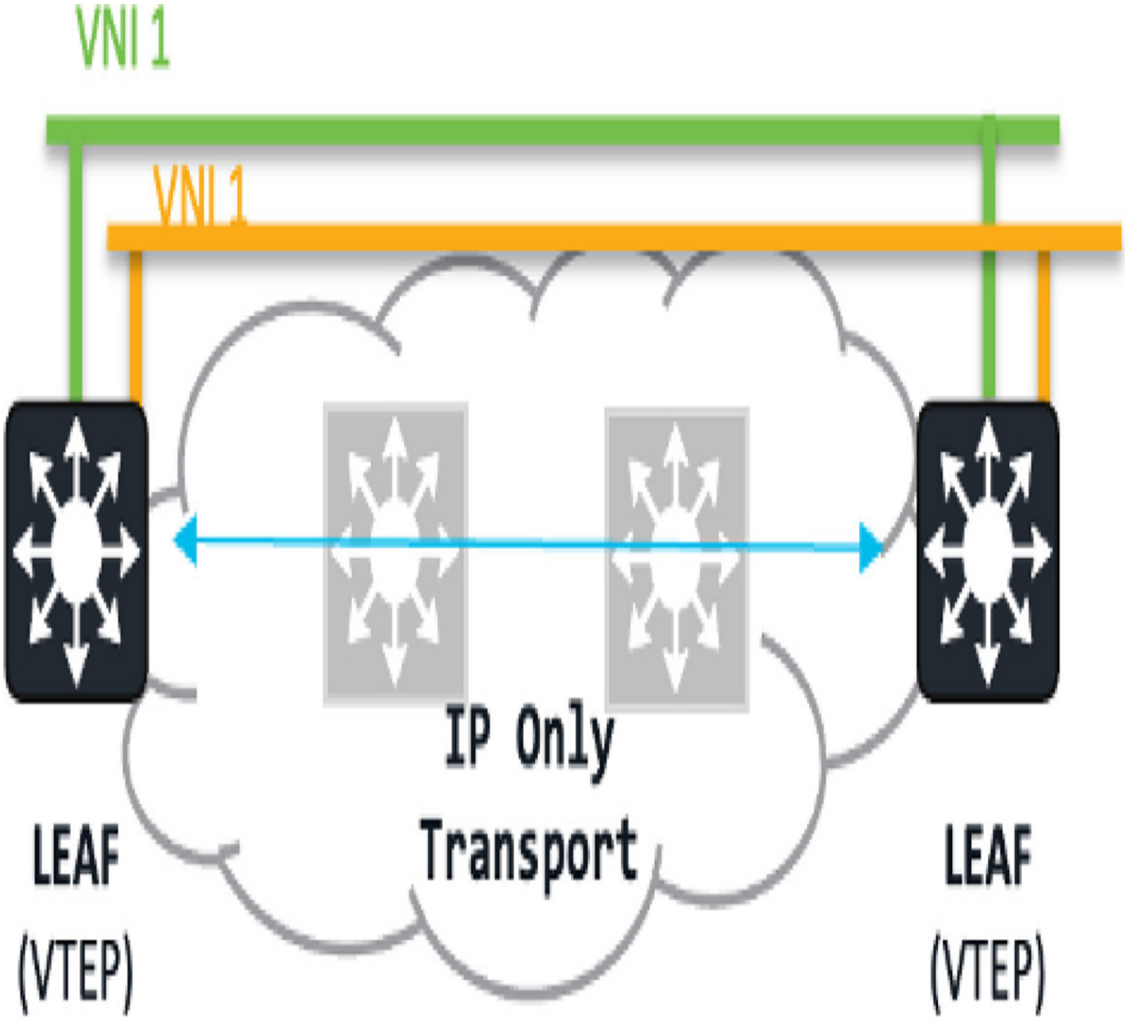


Figure 22-9 *VXLAN-EVPN Leaf-to-Leaf Architecture*

Leaf-to-leaf architecture can be particularly effective for policy propagation across larger segments of IP transport networks. It combines scalability with simplicity, making it a great option for use cases where you want to keep things straightforward without sacrificing performance.

By leveraging the flexibility of EVPN, you can design a network architecture that aligns perfectly with your operational needs, ensuring efficient policy propagation and robust performance across the board.

Security Considerations in the Campus

In recent years, the proliferation of network-connected devices has led to an exponential rise in security threats. These threats pose significant challenges to IT infrastructure, resulting in substantial costs related to both mitigation and recovery from various security incidents. Cyberattacks have grown more frequent, sophisticated, and varied. From ransomware and phishing to zero-day vulnerabilities, organizations are facing the constant challenge of increasingly complex attacks that can bypass traditional security mechanisms.

IT departments must continuously evolve and increasingly adopt advanced security measures to reduce potential impacts and gain deeper visibility into the traffic traversing their networks and their defenses to keep pace with these threats.

Particularly in campus environments, where diverse endpoints such as user devices, audiovisual systems, printers, sensors, and building management systems (BMSs) are connected, it is critical to implement security mechanisms as close as possible to these connection points. This approach not only strengthens traffic enforcement but also enhances telemetry and visibility. The data collected through these systems must be analyzed thoroughly to inform the development of effective policy enforcement strategies.

This growing complexity is one of the primary reasons why many organizations are integrating security components into their campus networks. Firewalls, including IPS, malware protection, threat intelligence feed, TLS/QUIC decryption, and encrypted visibility engine, play a pivotal role in mitigating a wide range of threats, particularly lateral threat propagation, where attackers move laterally across the network after gaining initial access. In both campus and large-scale network environments, preventing lateral movement is essential for containing security breaches and minimizing their impact.

Historically, firewalls served as the primary security perimeter for networks, regulating incoming and outgoing traffic—primarily north-south communication—based on predefined security policies. They provided organizations with protection against external threats to internal systems. However, in the context of a campus environment, the need to secure a wide variety of endpoints presents unique challenges. A key vulnerability arises from the differing levels of native security built into these various endpoints. For example, printers, which often reside within the same network segment as workstations, typically lack advanced security features. This exposes them as potential weak points, introducing considerable risk to other connected devices. The absence of sophisticated security mechanisms in such endpoints amplifies the need for robust network defenses to safeguard against potential breaches.

Taking this into account, organizations are developing their security policies with the main principle of placing potentially vulnerable endpoints into separate segments and protecting the east-west-communication through stateful inspection mechanisms.

Firewall Connectivity in the Campus

A traditional campus segmentation model typically utilizes VLAN-based mechanisms, wherein various types of endpoints are assigned to distinct VLANs. While this approach offers a certain degree of network segmentation, it constrains the effectiveness of security tools in conducting comprehensive traffic inspection and enforcement, both intra- and inter-VLANs. Consequently, this approach has proven inadequate in addressing modern threats and achieving the overarching objective of isolating various client types within the campus network.

In contrast, a macrosegmentation model based on virtual routing and forwarding (VRF) necessitates a fundamentally different approach to network design. This approach typically involves a comprehensive transition to routed-access architectures, wherein Layer 3 (L3) segments are extended down to the access layer, allowing for full segmentation across both Layer 2 (L2) and Layer 3 (L3).

In a typical campus three-layer architecture, a decentralized security component is integrated at the distribution or core layers, where north-south traffic undergoes inspection. This inspection can occur either inline, with all traffic passing through the security devices, or via an "on-a-stick" configuration, where traffic is selectively redirected for inspection. The redirection is implemented through either policy-based routing (PBR) or routing table decisions correlated with the virtual routing and forwarding (VRF) instance. East-west communication within a given segment (VLAN) is locally switched at the access layer, while inter-segment traffic is typically routed at the distribution or core layer. However, this routing can be modified through policy-based routing, allowing traffic to be redirected to the firewall for inspection.

When the security policy strictly prohibits communication between segments, such as IoT and employee networks, it introduces challenges related to maintaining complex routing configurations at the distribution or core layer. This complexity arises from the need to selectively redirect traffic for further inspection by the firewall. An alternative approach involves extending the L2 segment to the firewall layer, ensuring that traffic remains fully isolated until it reaches the firewall, where the default gateway for the specific segment is typically configured. While this method, also shown in Figure 22-10, offers complete isolation, it does not provide exceptions to the principle that all traffic must traverse the firewall.

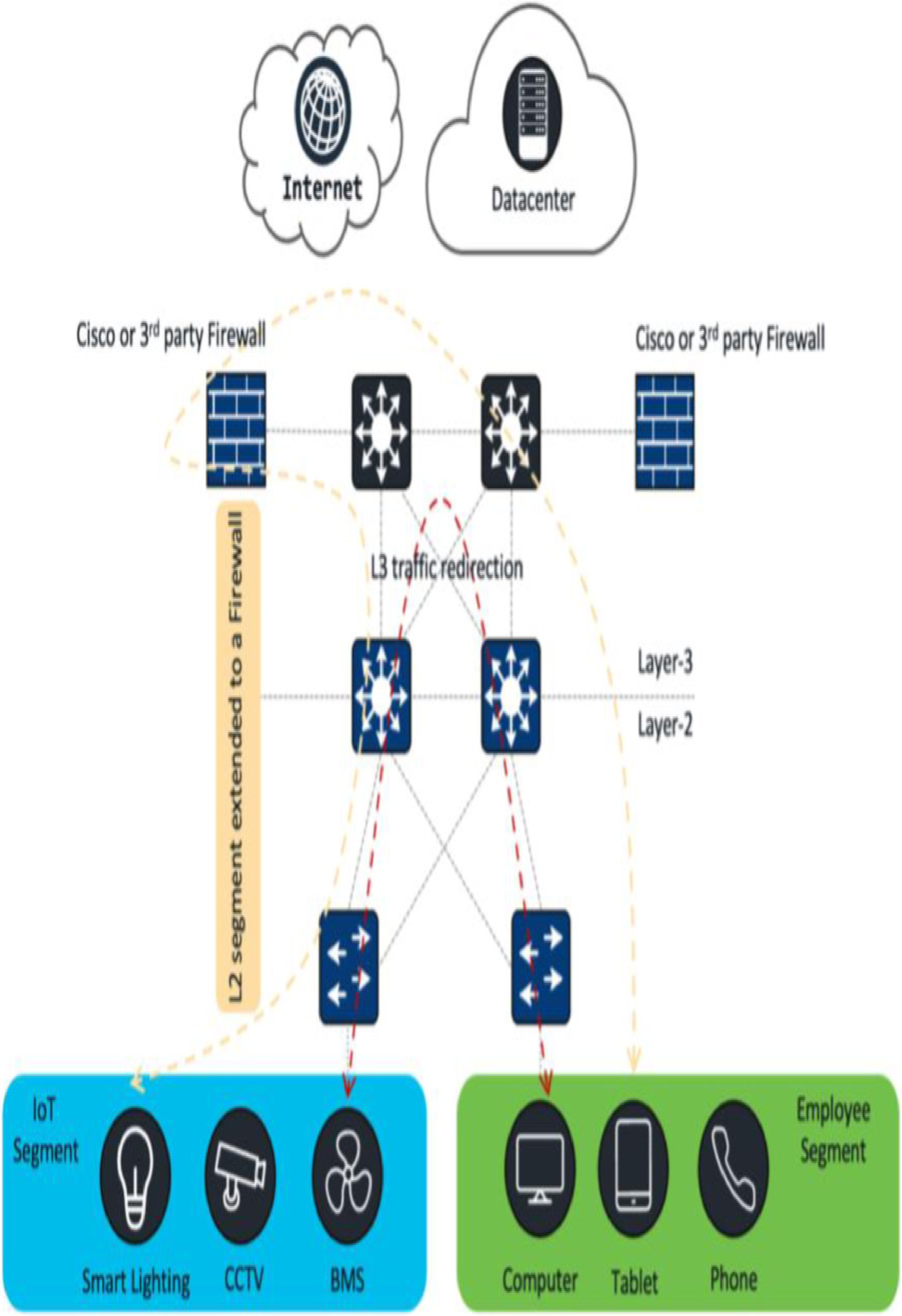

Internet
Datacenter
Cisco or 3rd party Firewall
Cisco or 3rd party Firewall
L2 segment extended to a Firewall
L3 traffic redirection
Layer-3
Layer-2
IoT Segment
Smart Lighting
CCTV
BMS
Employee Segment
Computer
Tablet
Phone

Figure 22-10 *Firewall Connectivity in Traditional Campus*

Campus networks based on underlay/overlay architectures, such as SD-Access, offer enhanced capabilities for segmenting and controlling specific types of traffic within the network fabric, eliminating the need for redirection to a firewall. These capabilities are built on Cisco TrustSec (CTS) and security group tagging (SGT) concepts. While this approach, shown in Figure 22-11, simplifies network design, it often fails to meet organizations' stringent requirements for stateful inspection between different segments, such as IoT and employee networks. The firewall continues to be a critical component of network security and requires tight integration to address organizational requirements effectively.

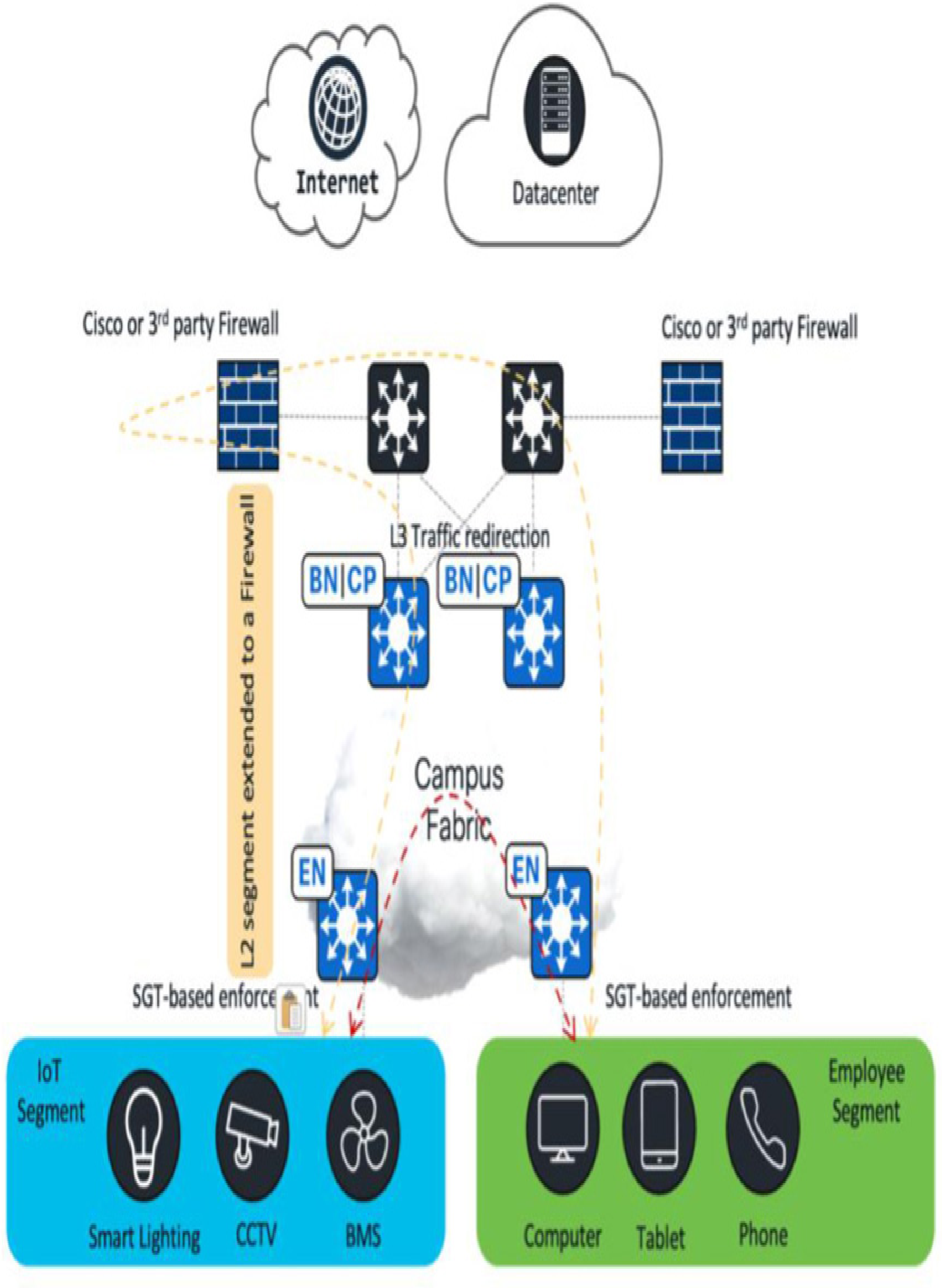


Figure 22-11 *Firewall Connectivity in SD-Access*

Within numerous organizations, there is a growing interest in integrating information technology (IT) and operational technology (OT) environments to achieve a more efficient operational framework through unified automation and orchestration solutions. However, this integration introduces certain risks, particularly concerning security.

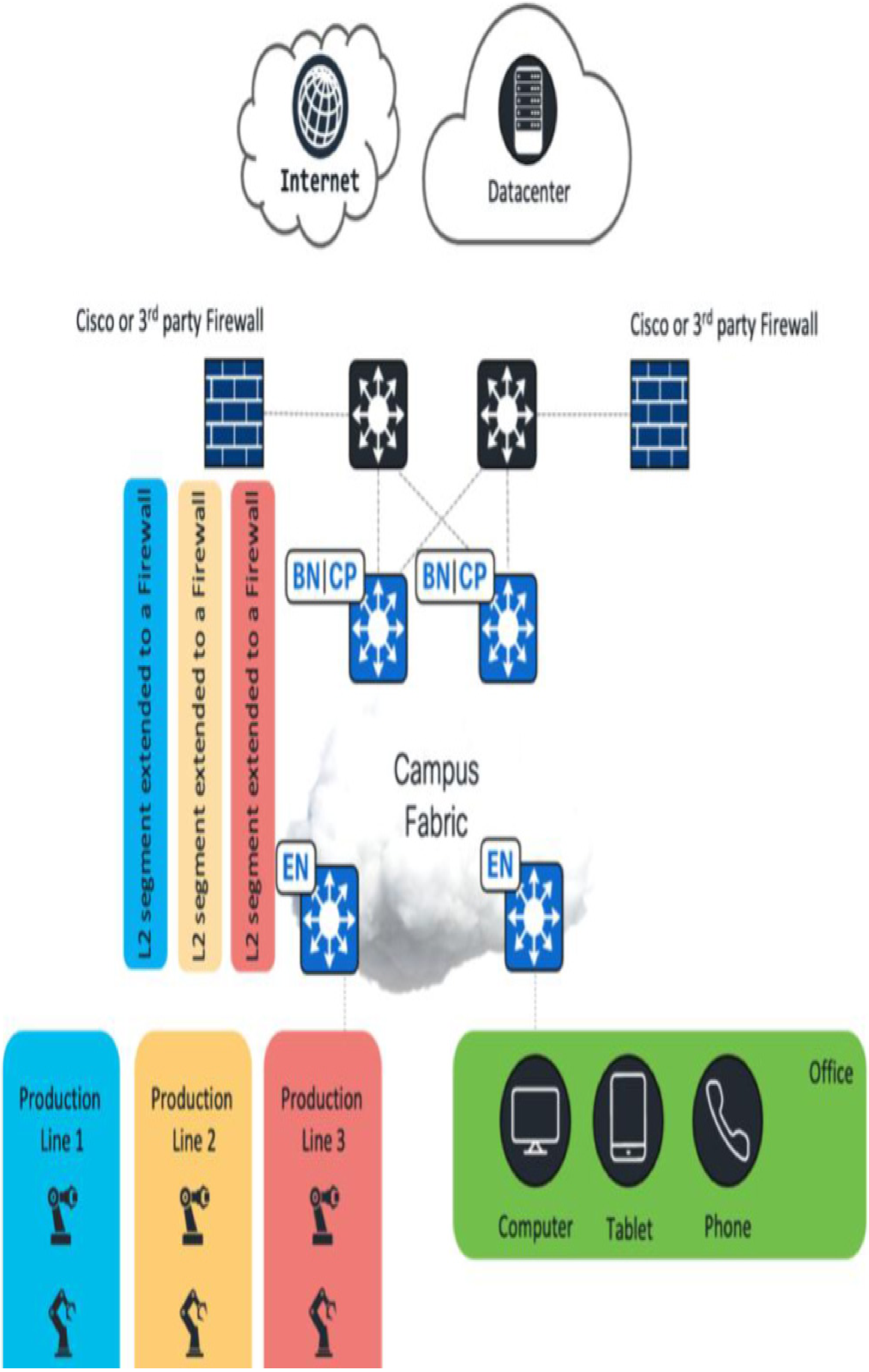
The SD-Access solution provides capabilities of integrating the fabric with Cisco or third-party vendor firewalls, regardless of whether the traffic is supposed to be L3 or L2 handed off from the Border nodes.

Consider an example, depicted in Figure 22-12, of a production/office campus site where you can consider two use cases:

1. Production line segment with full isolation and east-west and north-south traffic communication only over the firewall
2. Office endpoints where traffic destined toward the Internet should be subject to the firewall inspection

By utilizing Catalyst Center automation, both use cases can be effectively addressed. For the first scenario, the approach involves deploying a dedicated L2 segment within the fabric site, assigning all production line endpoints to this segment. The built-in workflow configures only an L2 VLAN and L2 virtual network identifier (L2VNID) as part of the LISP configuration, without establishing a default gateway on any fabric device. In these types of deployments, the default gateway will be configured outside of the fabric, and the fabric will act as an L2 transport only. Additionally, Catalyst Center enables Layer 2 flooding, a functionality necessary to propagate L2 traffic (particularly broadcast traffic) across the fabric.

To deploy an L2 segment using the L2 flooding feature, the underlay must be configured with appropriate multicast settings. Finally, the solution requires traffic handoff from the Fabric Border nodes to the firewall. On the firewall side, the necessary configuration involves defining the segment and establishing the default gateway. This setup, depicted in Figure 22-12, known as the "Gateway Outside of the Fabric," utilizes the SD-Access fabric solely as a Layer 2 transport.


Internet
Datacenter
Cisco or 3rd party Firewall
Cisco or 3rd party Firewall
L2 segment extended to a Firewall
L2 segment extended to a Firewall
L2 segment extended to a Firewall
BN|CP
BN|CP
Campus Fabric
EN
EN
Production Line 1
Production Line 2
Production Line 3
Office
Computer
Tablet
Phone

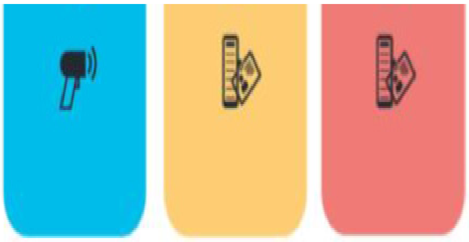


Figure 22-12 *Firewall with Gateway Outside of the Fabric in OT Vertical*

In the second scenario depicted in Figure 22-13, dedicated office endpoints will be associated with a specific virtual routing and forwarding (VRF) and virtual network (VN), utilizing a Layer 3 virtual network identifier (L3VNID) as part of the LISP construct. Traffic will be routed across the fabric until it reaches the Fabric Border nodes, where a Layer 3 handoff will be configured toward the upstream layer, along with BGP peering. The border devices will establish BGP peering with both the upstream (edge) layer, connecting to the broader network (WAN/DC), and with a set of firewall nodes. Traffic redirection decisions between the firewall and the campus edge will be made based on routing prefixes. More specific prefixes will be routed directly, while any unknown traffic will follow the default route directed toward the firewalls. The north-south traffic from the production lines will be filtered through the firewall stateful inspection while allowing the inter-L2 segment stateless enforcement leveraging Fabric TrustSec.

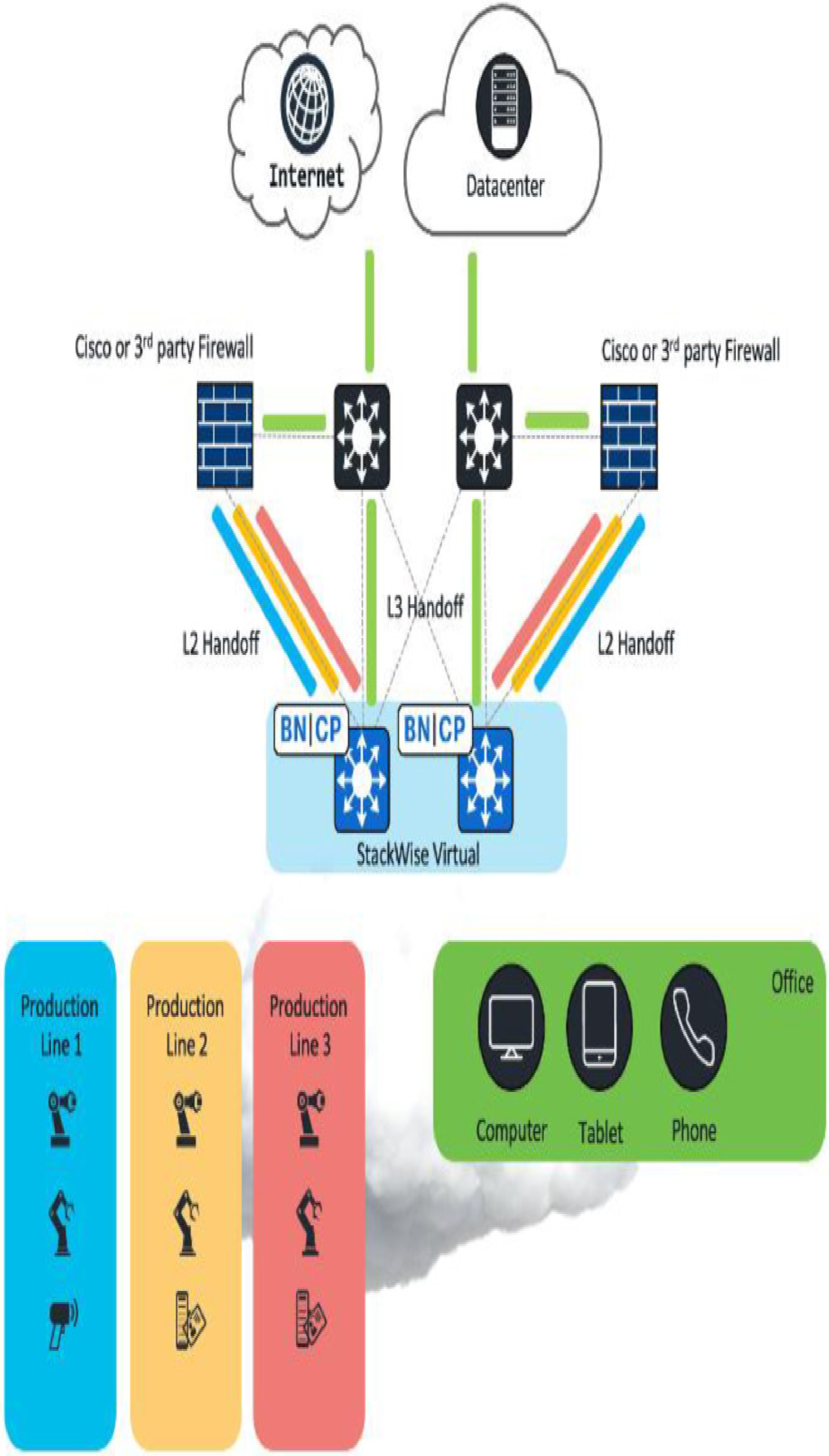


Figure 22-13 *Firewall L2 vs. L3 SD-Access Handoff Model*

A critical consideration in this design is the mechanism for L2 loop avoidance. Given that there is a simultaneous Layer 2 handoff to the external domain from the perspective of the SD-Access Fabric, it is essential to ensure that loops are effectively prevented. Any broadcast traffic handed off via a Border node must not return and be processed by another Border node; otherwise, due to the nature of the L2 virtual network identifier (L2VNID) within the fabric, it could be amplified, resulting in a broadcast storm. Implementing a dedicated interface for the L2 handoff facing the firewall should help mitigate this issue; however, careful planning is still required to avoid any potential loop-related challenges. The optimal scenario involves deploying the border function as a StackWise Virtual pair. StackWise Virtual is a technology that allows two switches to operate as a single logical switch. From a Spanning Tree perspective, this configuration operates as a single logical entity, thereby helping to mitigate potential Layer 2 issues.

More considerations related to the resiliency aspects of connecting the firewall will be covered in the upcoming section.

These scenarios present a fundamental challenge regarding scalability. In cases where an organization requires several hundred isolated spaces, particularly in an OT environment, the task of creating separate segments and extending them toward the firewall introduces significant complexities, particularly related to the scaling limitations of switches and firewall VRF or VLAN configurations.

Moreover, if flexibility is needed to create and remove these segments on an ad hoc basis, it results in considerable operational overhead and delays in provisioning, which do not align with the standards of software-defined networks (SDNs). Typically, not the entire traffic should be processed by security functions, but a lack of filtering forces all traffic to be sent to a firewall. This implies increased CapEx with an increase in the volume of traffic.

A potential solution to this challenge is the concept of Secure Service Insertion. It is a function whereby the SGT will be used to steer traffic toward a firewall and back and allows for the granular and dynamic selection of traffic destined for firewall inspection based on the classification of the source and destination traffic.

Secure Service Insertion provides

- **Separation of Roles:** You can establish clear distinctions between network operations (NetOps) and security operations (SecOps) roles. The security policy definition is a function of SecOps, while the steering policy definition falls under the purview of NetOps.
- **Vendor Agnostic Approach:** This capability allows the integration with firewall, IPS, or other security appliances, either from Cisco or third-party vendors.
- **Elastic Expansion:** You can facilitate the elastic expansion of security capacity in response to evolving requirements.
- **Utilization of Network Infrastructure:** You can leverage the existing network infrastructure to distribute security functions effectively.

The whole concept is based on the principle that there is an appropriate endpoint classification performed at the access layer, and based on the policy matrix, an appropriate traffic steering/policy enforcement is performed.

Consider an organization operating within a highly dynamic environment, managing numerous simultaneous projects, each requiring access to a distinct physical environment, as shown in Figure 22-14. A group of research and development contractors has been engaged to contribute to specific projects, with some contractors assigned to multiple projects simultaneously. According to the organization’s security policy, all communications initiated by contractors accessing their designated projects must undergo inspection by the firewall. Concurrently, communication between contractors and project workspaces to which they are not assigned must be strictly monitored and logged to ensure compliance with the security policy.

Deploying an environment with the set of the aforementioned requirements in the traditionally built campus network design would be a highly challenging exercise.

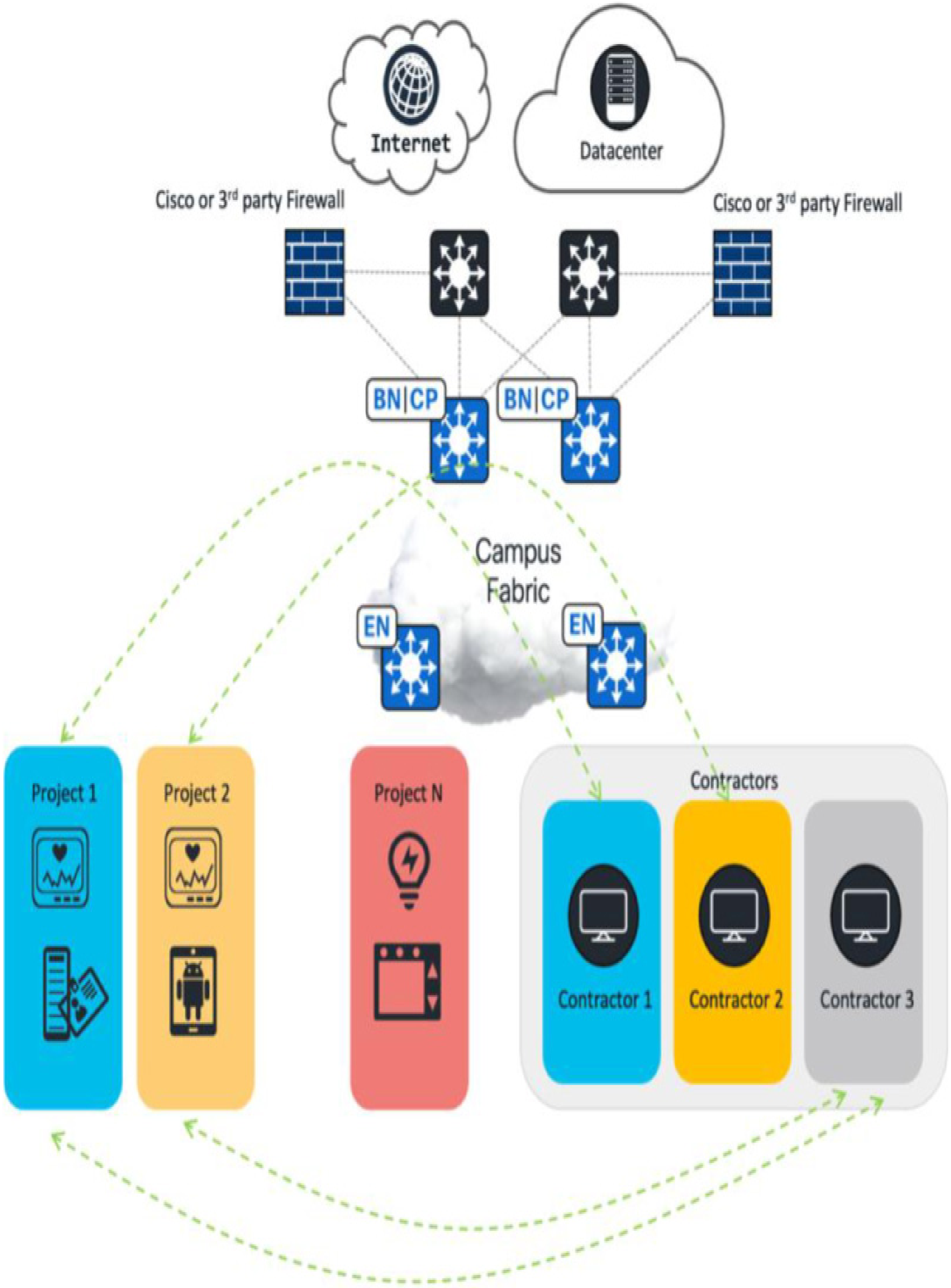
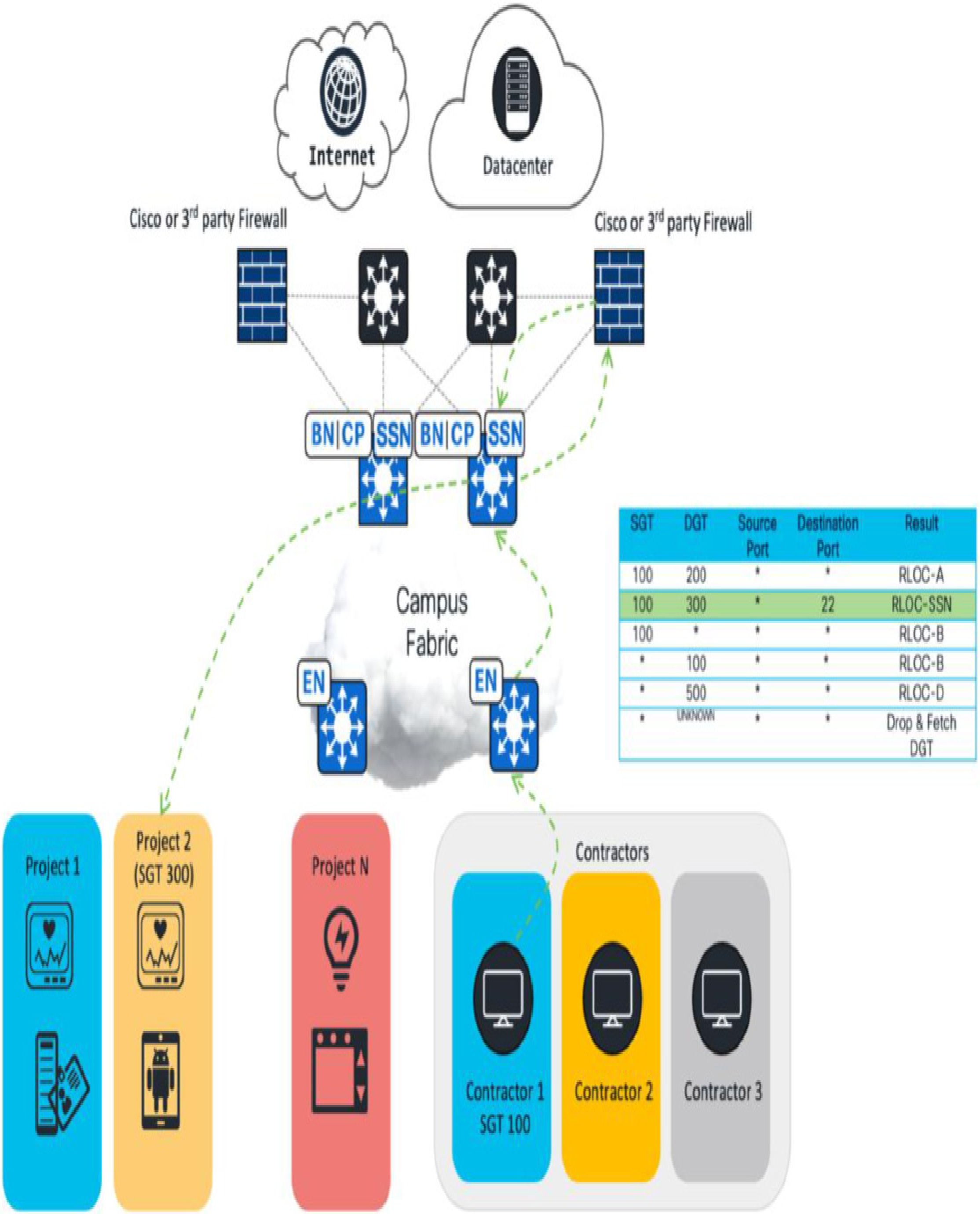


Figure 22-14 *R&D Environment*

Secure Service Insertion, as shown in Figure 22-15, introduces a policy-driven approach to traffic redirection at the access layer, enabling granular redirection of traffic to the firewall based on a combination of source and destination SGTs. To facilitate this, a new function called the Secure Service Node (SSN) integrates with either a Cisco or third-party firewall. Typically, the SSN is co-located on the border device.

In the context of the project-contractor association, traffic between contractors and any workload outside a designated campus site should be routed directly upstream. However, traffic between contractors and project endpoints should be subject to firewall inspection. When traffic reaches the Fabric Edge device, it passes through the forwarding engine, where one of the key steps involves a policy-based routing TCAM check, which may result in traffic redirection toward the SSN node.



SGT	DGT	Source Port	Destination Port	Result
100	200	*	*	RLOC-A
100	300	*	22	RLOC-SSN
100	*	*	*	RLOC-B
*	100	*	*	RLOC-B
*	500	*	*	RLOC-D
*	UNKNOWN	*	*	Drop & Fetch DGT

Figure 22-15 *R&D Environment and Secure Service Insertion*

Once the packet reaches the SSN node, it is directed to the appropriate interface toward the firewall for processing. After the firewall processes the traffic, it returns it to the SSN node. Upon receiving the return traffic, the SSN forwards it based on the LISP map cache.

This solution offers complete flexibility, including the ability to enhance firewall throughput by distributing specific security functions across the network. Additionally, it optimizes bandwidth usage by dropping unwanted traffic as close as possible to its source.

> **Disclaimer**
>
> As of the writing of this chapter, the functionality for selective traffic redirection to Cisco or third-party firewalls is available in non–SD-Access campus environments. This capability is expected to be implemented in the SD-Access solution in the near future.

Third-Party Vendor Firewall Policy Integration

In the previous sections, we discussed various use cases and considerations related to data plane traffic propagation toward the firewall, including when and how to utilize Layer 3 versus Layer 2 and the factors that may influence design decisions. While these elements are crucial, equally important is the full integration of the security components from a policy plane perspective.

Organizations are seeking effective solutions to securely manage access between users, endpoints, and applications across enterprise, Internet, and cloud environments. Achieving this requires a consistent approach to uniquely identify users, endpoints, and applications based on contextual information, which must be shared across domain controllers.

In response to this need, Cisco has introduced a framework called Common Policy. This framework enables Cisco products, along with third-party solutions, to share contextual data about users, endpoints, and applications, allowing organizations to create and enforce consistent access policies within their chosen domains. The integration is achieved by leveraging the PxGrid protocol that is designed to share contextual data and security information between various platforms and devices seamlessly.

Within the Common Policy framework, Cisco Identity Services Engine (ISE) plays a pivotal role in gathering application context from various on-premises and cloud components. It then shares this contextual information —related to users, devices (endpoints), and applications—across the broader ecosystem, as shown in Figure 22-16, including third-party devices. Key examples of shared context include security group tags, endpoint policy groups, or endpoint security groups (from the APIC Data Center), all of which are associated with IP addresses. This context is distributed across different domains.

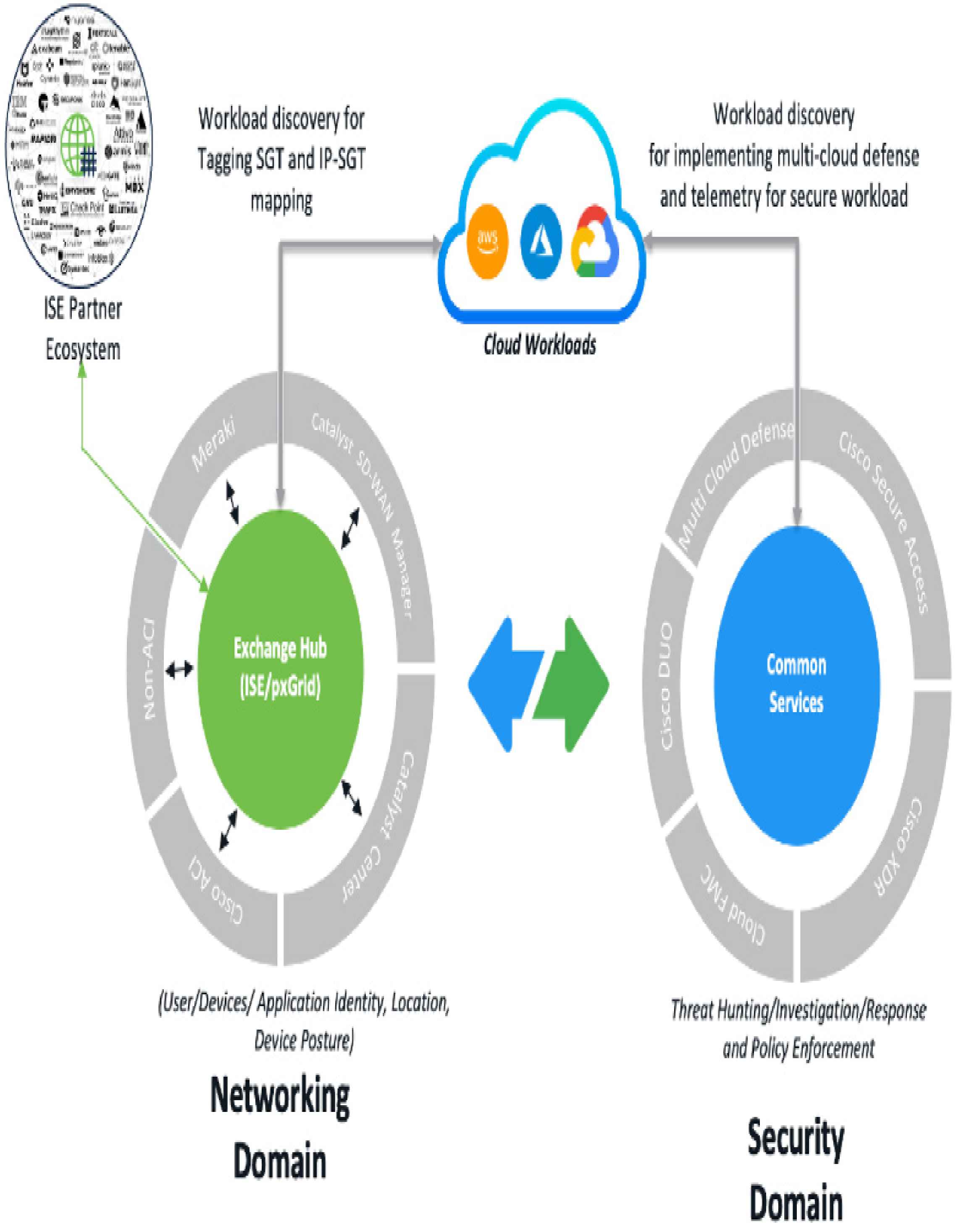
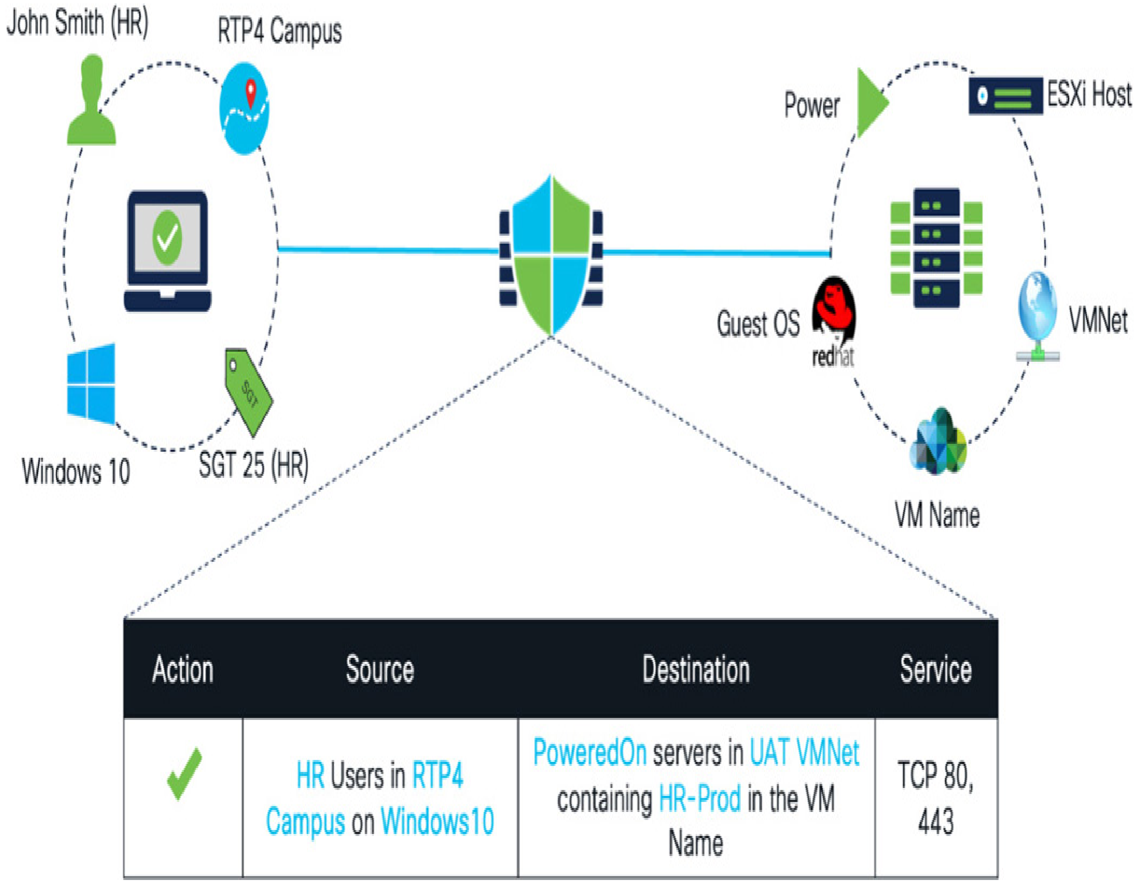

ISE Partner Ecosystem
Workload discovery for Tagging SGT and IP-SGT mapping
aws
Cloud Workloads
Workload discovery for implementing multi-cloud defense and telemetry for secure workload
Meraki
Catalyst SD-WAN Manager
Non-ACI
Exchange Hub (ISE/pxGrid)
Catalyst Center
Cisco ACI
(User/Devices/ Application Identity, Location, Device Posture)
Networking Domain
Multi Cloud Defense
Cisco Secure Access
Cisco DUO
Common Services
Cisco XDR
Cloud FMC
Threat Hunting/Investigation/Response and Policy Enforcement
Security Domain

Figure 22-16 *Networking and Security Domains Integration*

Cisco’s Secure Firewall Threat Defense systems have been supporting PxGrid integration for years, and other vendors are gradually adopting this mechanism. This tight integration across the entire security ecosystem provides more visibility and, what is more important, contextual policy-driven decisions that are particularly important in today’s rapidly evolving multivendor environments.

Traditionally, firewall rules have been based on Layer 3 and Layer 4 conditions. However, this approach is increasingly inadequate in modern network and application environments, where users and endpoints are no longer tied to fixed locations, and applications can be deployed in minutes in hybrid or cloud environments. As a result, network policies must adapt dynamically. Additionally, as noted by Gartner, "Through 2023, 99% of firewall breaches will be caused by misconfigurations, not by flaws in the firewalls themselves." (See the “References” section for more details.) This highlights the industry’s shift from an IP-based approach toward intent-based policies.

As depicted in Figure 22-17, modern firewall policies can now be based on certain combinations of conditions that are contextual and provide an IP-based agnostic approach focusing on the actual group belonging based on the source and destination attributes.



Action	Source	Destination	Service
✓	HR Users in RTP4 Campus on Windows10	PoweredOn servers in UAT VMNet containing HR-Prod in the VM Name	TCP 80, 443

Figure 22-17 *Context-Based Firewall Rules*

Consider an organization operating a dynamic environment with multiple R&D spaces and associated contractors. The goal is to ensure that, regardless of where these contractors access the network, they consistently have the same access privileges to the relevant R&D projects. Additionally, any traffic originating from the contractors' segment and directed toward these spaces must be inspected, monitored, and logged by the firewall. Furthermore, the organization follows a multivendor solution approach, utilizing third-party firewalls to meet these security requirements.

This approach presents a challenge in effectively managing firewall rules while maintaining location-independent policies. Traditionally, this effort would involve creating a separate L3 segment for contractors and configuring access rules for the relevant R&D spaces. However, if a contractor were to work from a different physical location, these rules would need to be updated accordingly. Such an approach is incompatible with the requirements of dynamic environments, which demand flexibility and agility while strictly adhering to security policies.

A more effective approach is to utilize the firewall's capabilities in defining SGT-based rules (shown in Figure 22-18), which are independent of IP addresses. This approach ensures that, regardless of the endpoint's network connection, it consistently receives the same tag. The IP-SGT mapping can then be leveraged by the firewall to make policy decisions. Whenever possible, source SGT propagation should be implemented through the data plane, eliminating the need for control plane propagation. This approach is increasingly being adopted in third-party security solutions, where the firewall can directly extract the source SGT from the packet in transit. However, it remains necessary to provide destination SGT information to the firewall.

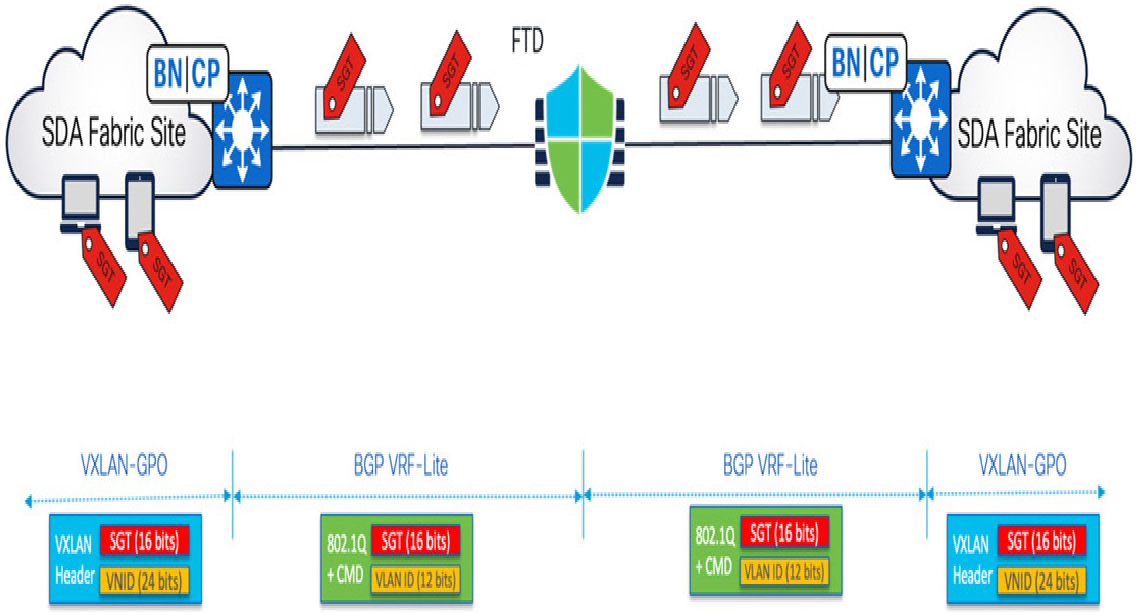
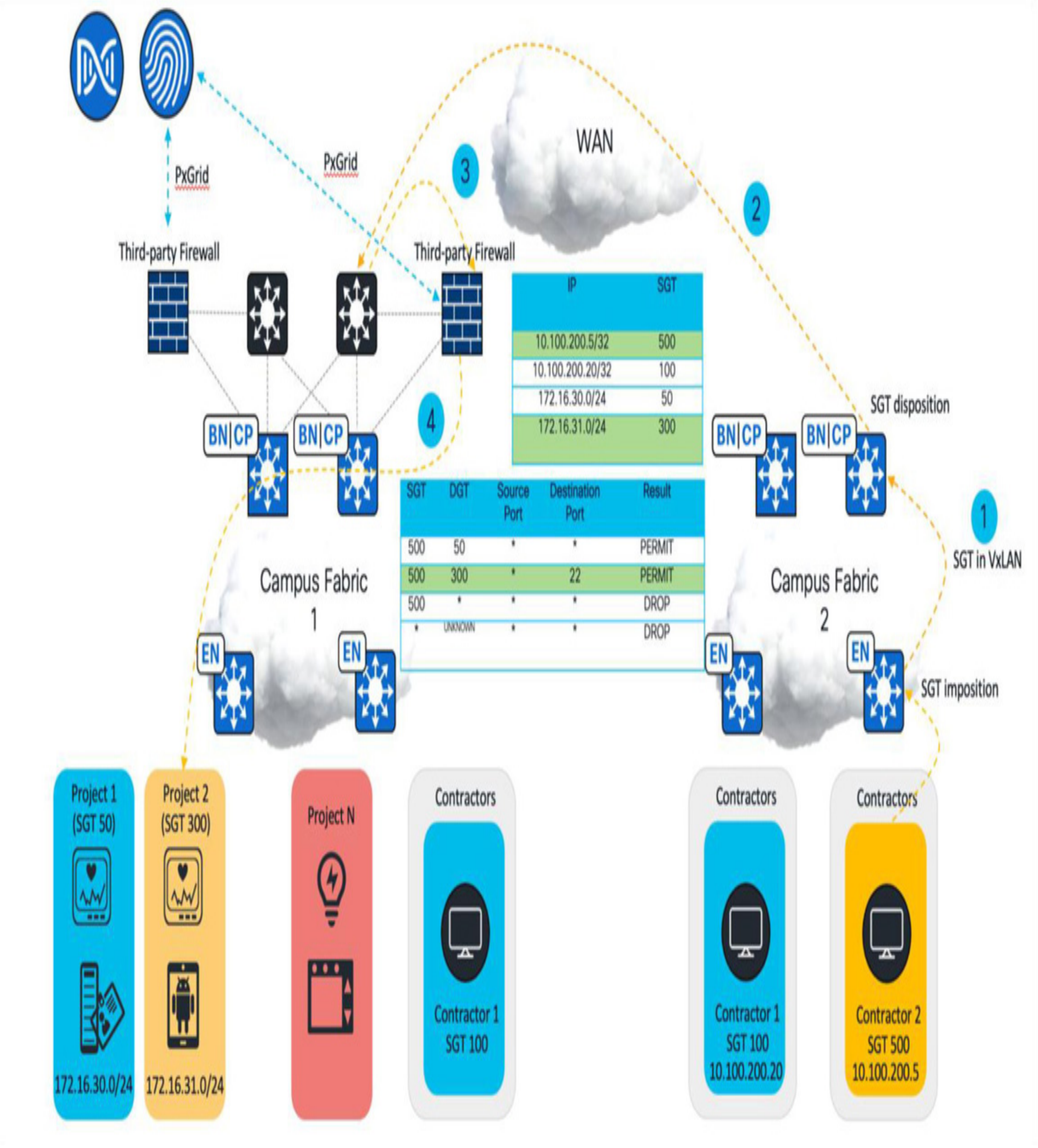


Figure 22-18 *CMD Inline Tagging SGT Propagation Across Firewall*

In cases where an organization has adopted a multivendor strategy and requires the use of third-party security components, integration with ISE can be achieved via PxGrid. In this setup, ISE functions as the PxGrid publisher, while the third-party firewall operates as the PxGrid subscriber. As ISE oversees endpoint authentication and authorization, it dynamically assigns network segments and SGTs based on the user's Active Directory group membership. Consequently, the IP-SGT mapping table is continuously updated within ISE. Furthermore, ISE maintains information regarding project-related devices or endpoints, whether learned dynamically or statically assigned.

Consider the scenario (see Figure 22-19) where contractors are connected to two separate fabric sites, and traffic is destined from Site 2 to projects hosted at Site 1. As the packet traverses Site 2 (Step 1), the Fabric Edge switch imposes a security group tag that is encapsulated in the VxLAN header. Upon leaving the border (Step 2), the SGT is removed, and the packet is forwarded over the WAN without the tag until it arrives at Site 1. Due to routing decisions, traffic directed toward the project networks is sent to the firewall (Step 3).

At this point, the firewall cannot extract source SGT data from the data plane (removed by the border at Site 2). However, when integrated with ISE via PxGrid, the source, destination SGTs, and IP-SGT mapping tables are shared, enabling the firewall to utilize this information for policy enforcement. For the firewall to make an informed policy decision, it requires both the source and destination SGT-to-IP mappings.



IP	SGT
10.100.200.5/32	500
10.100.200.20/32	100
172.16.30.0/24	50
172.16.31.0/24	300

SGT	DGT	Source Port	Destination Port	Result
500	50	*	*	PERMIT
500	300	*	22	PERMIT
500	*	*	*	DROP
*	UNKNOWN	*	*	DROP

Figure 22-19 *Third-Party Vendor Firewall Integration Packet Flow*

Cisco has already implemented SGT-based policies in its products, and many other vendors are gradually adopting this approach in their products and solutions.

Cisco's Secure Firewall Threat Defense can derive the source SGT from the CMD header as well as integrate with the Identity Services Engine to obtain the destination SGT along with the IP-SGT mapping table. All the data then can be leveraged as a condition in the Attribute-Based Policy, including near-real-time object updates.

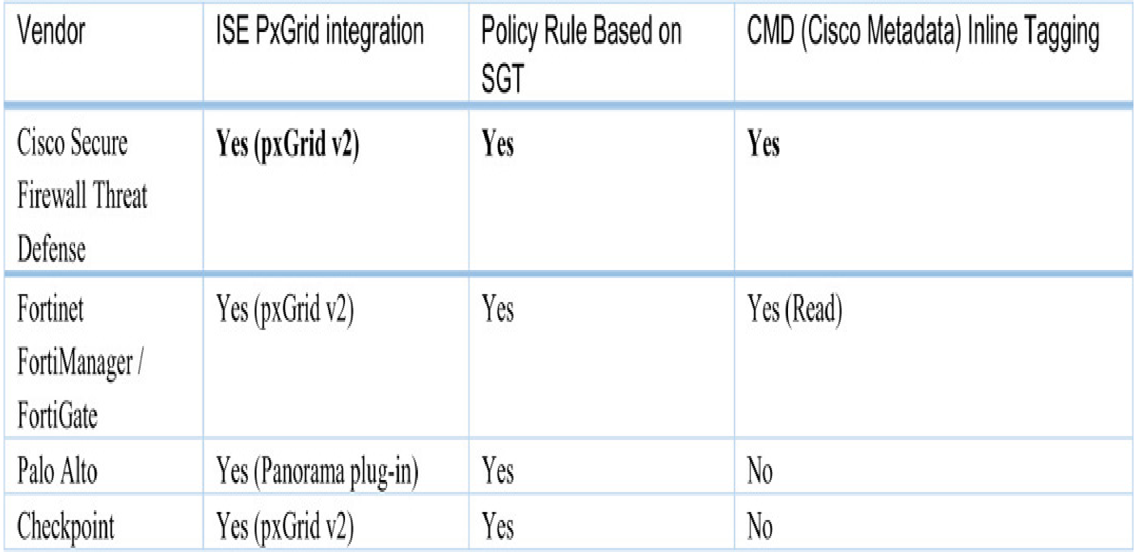
When it comes to other vendors, Palo Alto Networks utilizes an additional plug-in called Panorama for Cisco TrustSec, which facilitates the creation of security policies. This plug-in monitors changes in TrustSec security groups and registers this information with Panorama. It forwards IP data to the firewall, so Panorama can apply the correct policy to corresponding endpoints. The plug-in is capable of supporting up to 16 Cisco ISE servers, ensuring robust scalability and integration.

Fortinet has similarly integrated the capability to work with Cisco ISE through PxGrid within its FortiGate firewall solutions. Additionally, FortiGate firewalls can read security group tags embedded in Ethernet frames (Cisco Metadata header) and use them as criteria for matching in firewall policies.

Checkpoint published a comprehensive white paper on integrating its Checkpoint Identity Collector with Cisco ISE. Per this white paper, “Cisco ISE integrates with Check Point’s software blade to provide real-time and comprehensive identity and network privilege context. That includes user IP address, name, group, and Cisco TrustSec® security group tag information.” (See the “References” section for more details.)

Table 22-1 presents the current set of SGT/PxGrid capabilities available across third-party vendors. For a detailed list of capabilities, consult the respective vendor’s documentation.

Table 22-1 *Multivendor PxGrid and SGT Capabilities*



Vendor	ISE PxGrid integration	Policy Rule Based on SGT	CMD (Cisco Metadata) Inline Tagging
Cisco Secure Firewall Threat Defense	**Yes (pxGrid v2)**	**Yes**	**Yes**
Fortinet FortiManager / FortiGate	Yes (pxGrid v2)	Yes	Yes (Read)
Palo Alto	Yes (Panorama plug-in)	Yes	No
Checkpoint	Yes (pxGrid v2)	Yes	No

Highly Resilient Firewall Integrations

Modern software-defined networks need to be both highly resilient and scalable. In the design of campus networks, a critical consideration is enhancing network convergence and ensuring protection against link or node failures. This core principle is embedded in best practices, focusing on introducing redundancy and the proper interconnection of each layer within the hierarchical campus design.

This approach should also be applied when designing interconnections to upstream layers, including the integration of third-party security components. However, this is often overlooked, despite its crucial role, because a single weak point in the design can disrupt overall network convergence.

When third-party firewalls are integrated, the interconnection methods can vary depending on the vendor's failover and clustering capabilities. Nevertheless, several common elements must be highlighted and taken into consideration.

Cisco's SD-Access Solution Design Guide (CVD) outlines and compares two common interconnectivity models: the square and triangle topologies. According to the guide, optimal network convergence is typically achieved by utilizing the triangle model, which maximizes the benefits of equal-cost multipathing.

Here, we will provide a detailed comparison of both topologies in the context of third-party firewall integration and analyze their impact on overall network convergence. The most common firewall deployment is an Active-Standby configuration, where only one physical node actively forwards traffic. When it is combined with the square design (see Figure 22-20)—where only one Fabric Border node connects to the active unit—additional configuration is required to fully utilize Fabric Border resiliency. This design is typically chosen when firewall units are located in separate locations, and physical connectivity constraints or limitations need to be considered. The setup requires extending the L2 segment across both firewall nodes (Active Firewall – Fabric Border 1 – Fabric Border 2 – Standby Firewall). From the perspective of the L3 handoff at the Fabric Border, a shared /29 subnet will be allocated to this segment, with each device allocated a unique IP address (two Fabric Border nodes and Firewall Virtual IP, or VIP, address). This enables the establishment of independent BGP sessions between both Fabric Border nodes and the active firewall unit. Both the upstream and downstream traffic will be load-balanced between two Border nodes. The upstream from the Fabric perspective will leverage the LISP Pub-Sub Dynamic Default Border functionality, whereas the downstream will follow the BGP route (assuming maximum paths or equivalent parameters have been configured on a firewall).

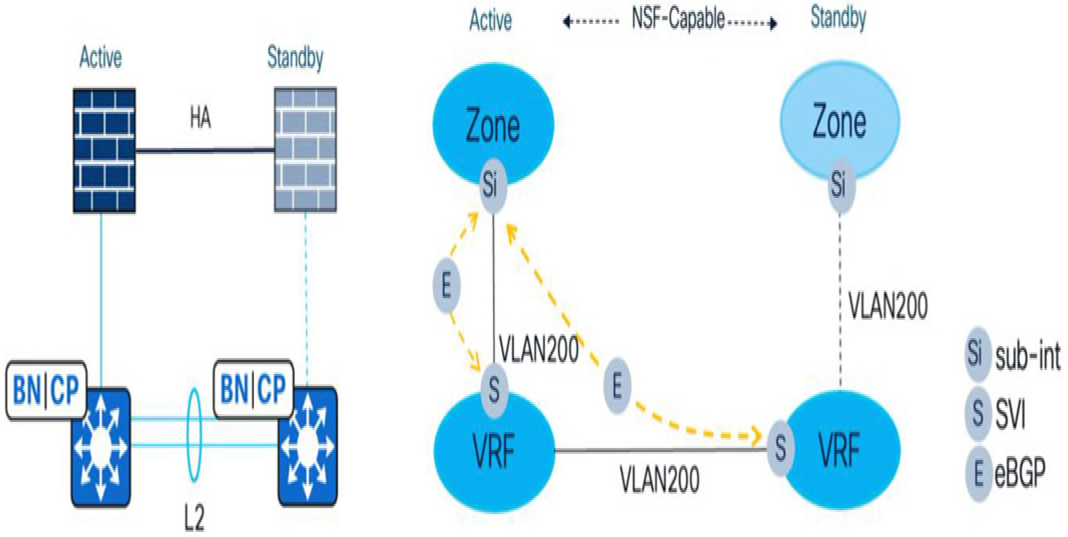


Figure 22-20 *Firewall and Fabric Border Square Design*

In the event of a firewall failover (see Figure 22-21), assuming that graceful restart (GR) and non-stop forwarding (NSF) mechanisms are configured, convergence should be decent. Depending on the firewall's failure detection mechanism, the process may take a few seconds, followed by the switchover of the virtual IP (VIP). From the perspective of the Fabric Border, no changes occur, because the VIP used to establish the BGP session with the Border nodes remains the same. A Gratuitous ARP (GARP) is expected to be generated to update the ARP table on the Border. Both NSF and GR mechanisms should ensure that the BGP session remains intact and does not terminate during the failover.

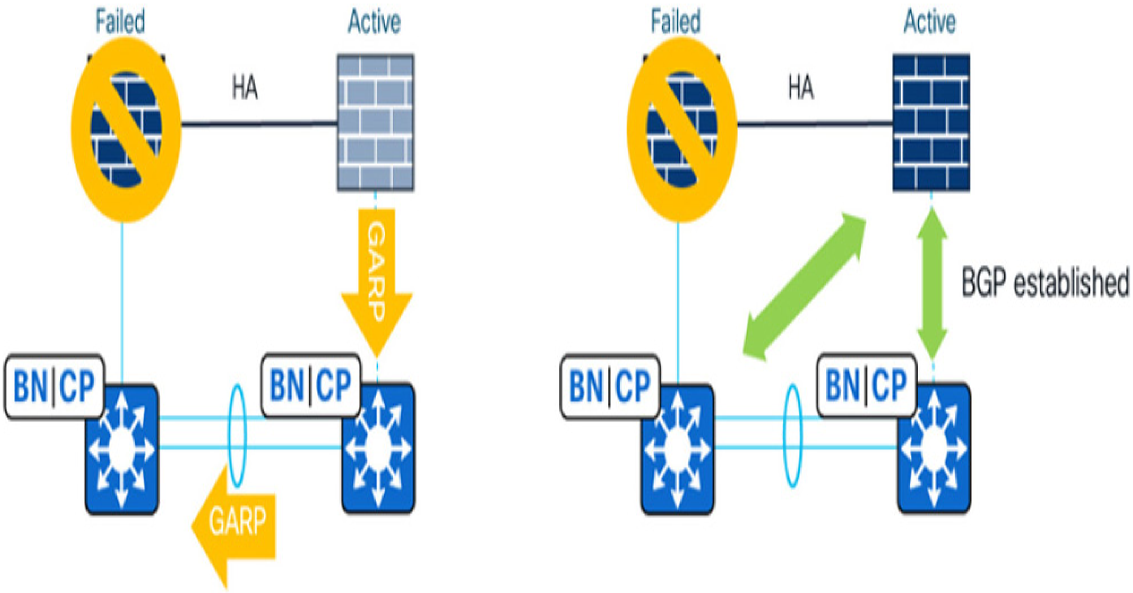


Figure 22-21 *Firewall Node Failover Scenario*

One challenge to consider is the potential delay in failure detection. In the event of an inter-border link or a Border node connected to a Standby firewall failure (see Figure 22-22), the firewall may take a significant amount of time to detect the change, particularly if it relies on default BGP timers. During this period, approximately half of the downstream north-south traffic could be blackholed because the active firewall remains unaware of the downstream failure. Consequently, during the inter-border link failure, approximately 50 percent of the upstream traffic from the Fabric Edge layer will also be impacted.

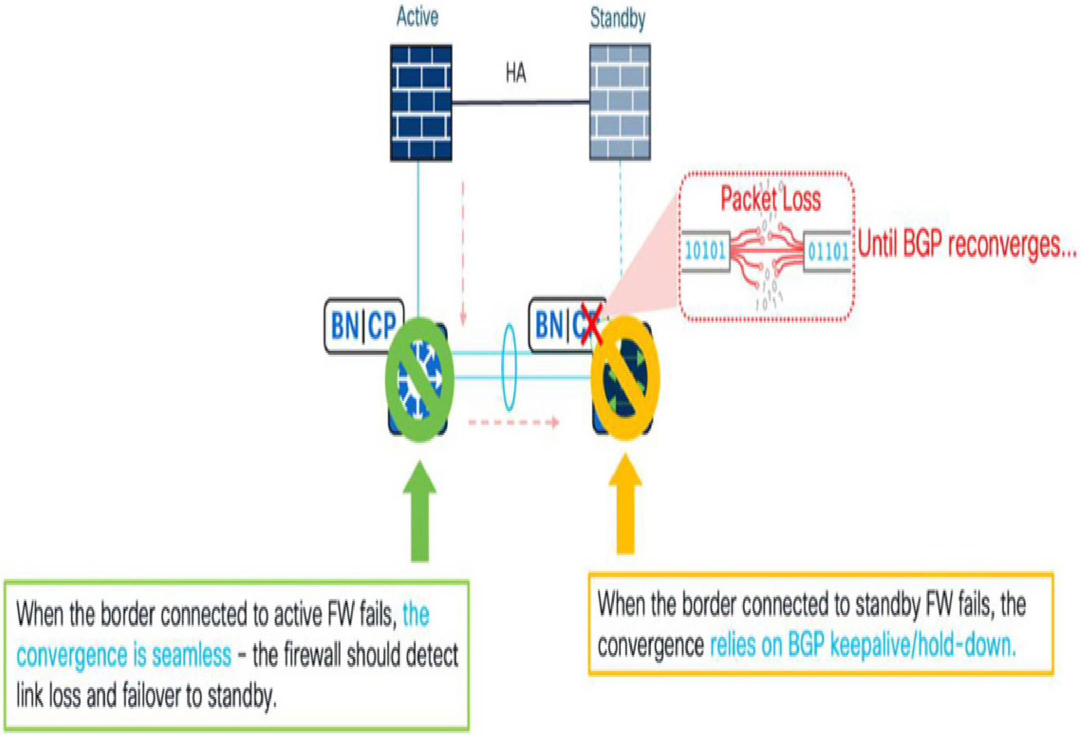

Active
Standby
HA
Packet Loss
10101
01101
Until BGP reconverges...
BN|CP
BN|CP
When the border connected to active FW fails, the convergence is seamless - the firewall should detect link loss and failover to standby.
When the border connected to standby FW fails, the convergence relies on BGP keepalive/hold-down.

Figure 22-22 *Inter-Border Link and/or Fabric Border Failure Scenario*

A potential solution is to reduce the BGP timers or implement bidirectional forwarding detection (BFD), which would significantly improve convergence due to its subsecond failure detection capabilities. However, a thorough analysis is required to assess the interoperability of BFD with NSF/GR. Typically, BFD may detect a failure more quickly and terminate the session, while NSF could still be in the process of completing the failover. These considerations should be carefully evaluated on a per-vendor basis, because third-party vendors may implement different failover mechanisms.

A considerably more optimal solution is to leverage a triangle approach (see Figure 22-23), assuming the physical connectivity is not a constraint. In such a scenario, each Border node has an individual physical and logical connection to the active and standby firewall units. Thanks to this design, the topology fully leverages the equal-cost multipathing (ECMP). Compared to the previous example, there is much more optimal traffic flow, eliminating the use of the inter-border link. The key consideration in such an approach is the asymmetrical traffic that is expected to be arriving at the firewall and the mechanism to deal with that.

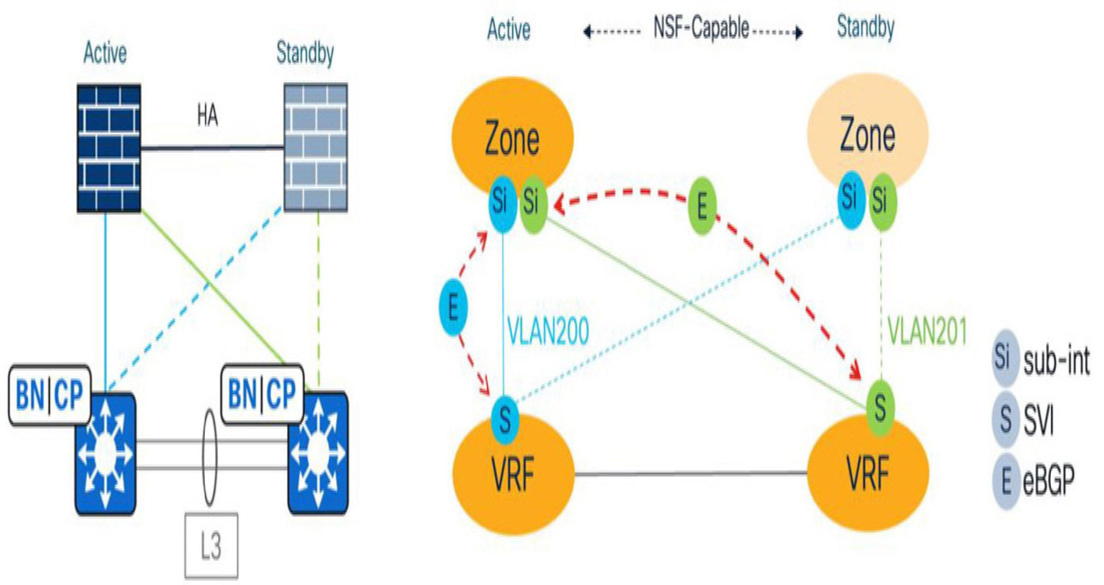


Figure 22-23 *Firewall and Fabric Border Triangle Design*

In the event of Border node failures (see Figure 22-24), convergence should occur within subsecond timescales because the firewall will detect the link failure to the affected Border node and promptly remove it from the routing table. The need for BGP timers or BFD is minimal because the link failure will automatically trigger the BGP session to go down, followed by the withdrawal of the affected routes, except in rare cases of indirect physical failure scenarios. While employing BFD may still be considered, as noted in the previous example, it could potentially cause more disruption than benefits, particularly by terminating the BGP session during a firewall node NSF failover.

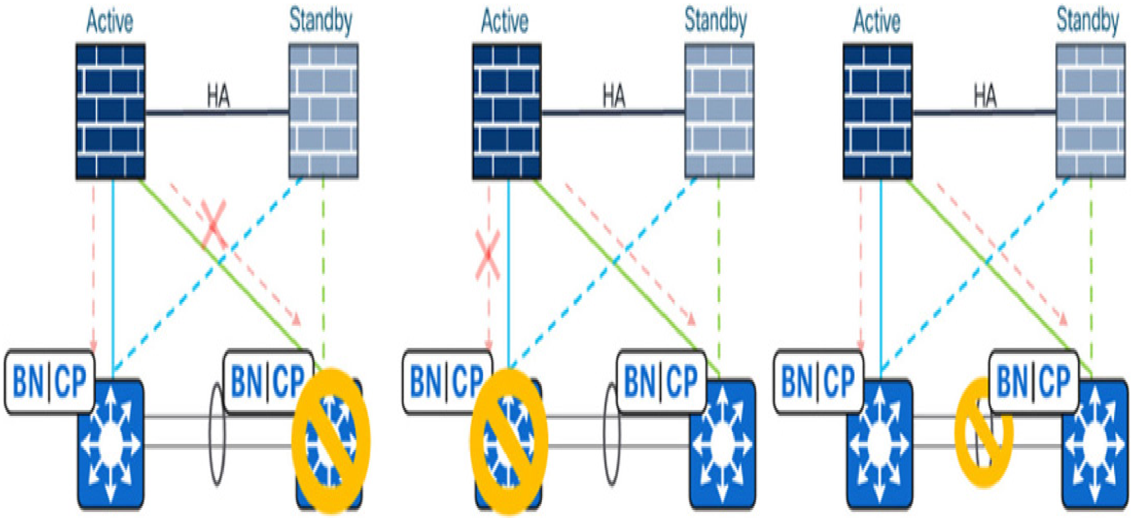


Figure 22-24 *Failure Scenarios in the Triangle Design*

An additional option for interconnecting both layers is clustering at both the Border and Firewall layers (see Figure 22-25). In this configuration, all firewalls within the cluster share a common interface IP address and routing process. However, this approach also requires the connected switches to appear as a single logical unit, typically using a Spanned EtherChannel. This can be achieved through Cisco StackWise Virtual (SWV) technology, where two physical devices are presented as a single logical node to neighboring devices. This approach significantly simplifies configuration while enhancing both resiliency and scalability. The switching mechanism remains based on non-stop forwarding (NSF) with graceful restart (GR).

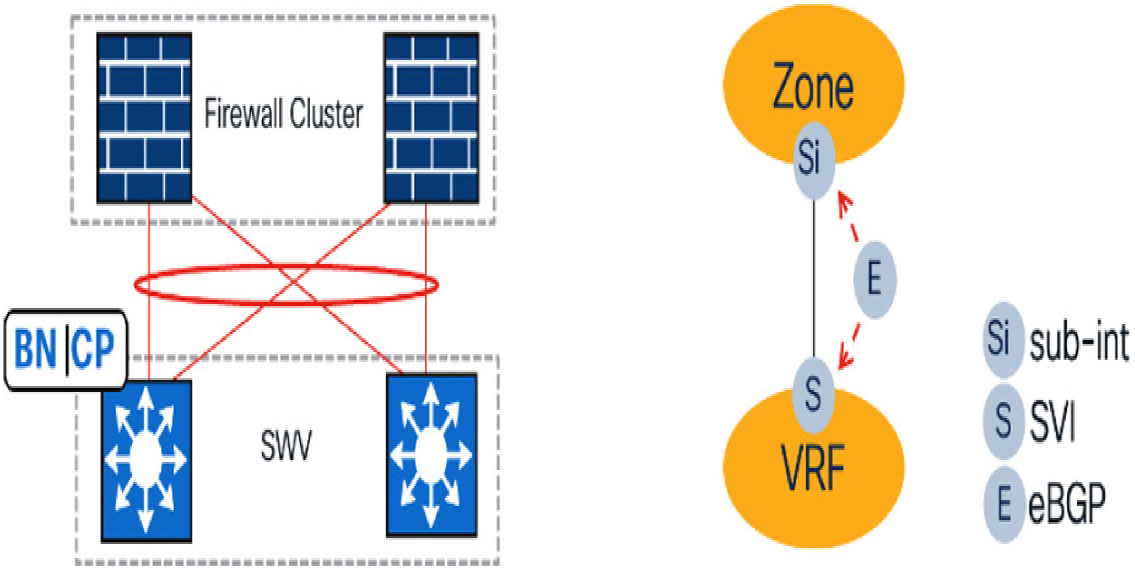


Figure 22-25 *Firewall Cluster Connectivity to StackWise Virtual*

Summary

Security threats pose significant challenges to any organization, resulting in substantial costs related to both mitigation and recovery from various security incidents. Due to its rapid growth, where businesses are heavily impacted across many verticals, zero trust architecture is becoming one of the key campus design principles. The organizations adopting the methodology are looking for an efficient integration of their nonhomogeneous infrastructures. Thanks to the wide market adoption of different integration mechanisms, having a multivendor organization strategy is no longer a show-stopper preventing the organization from implementing the end-to-end policy strategy. This strategy includes providing a consistent, informed, and context-aware methodology where every device connected to the network can be uniquely identified by its function and the traffic appropriately classified and enforced within the domain of choice.

References

1. Gartner Research, *Technology Insight for Network Security Policy Management*, February 2019: https://www.gartner.com/en/documents/3902564/technology-insight-for-network-security-policy-management
2. Palo Alto, Static Security Group Tag (SGT) for TrustSec Plugin: https://docs.paloaltonetworks.com/pan-os/11-0/pan-os-new-features/panorama-features/static-security-group-tag-sgt-support-for-trustsec-plugin
3. Fortinet, Cisco Security Group Tag as Policy Matching Criteria: https://docs.fortinet.com/document/fortigate/7.0.0/new-features/322202/cisco-security-group-tag-as-policy-matching-criteria-7-0-1
4. Check Point, Check Point and ISE Integration White Paper: https://community.checkpoint.com/fyrhh23835/attachments/fyrhh23835/general-

topics/10644/1/Check%20Point%20and%20ISE%20Intergration%20White%20Paper.pdf

5. Campus LAN and Wireless LAN Solution Design Guide: https://www.cisco.com/c/en/us/td/docs/solutions/CVD/Campus/cisco-campus-lan-wlan-design-guide.xhtml
6. Cisco SD-Access Solution Design Guide (CVD): https://www.cisco.com/c/en/us/td/docs/solutions/CVD/Campus/cisco-sda-design-guide.xhtml#Layer3RoutedAccessandSDAccessNetworkDesign
7. BGP Maximum-paths: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9600/software/release/17-12/configuration_guide/mpls/b_1712_mpls_9600_cg/configuring_eibgp___multipath.xhtml
8. BGP NSF Awareness: https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/ip-routing/b-ip-routing/m_irg-nsf-awareness.xhtml
9. Cisco StackWise Virtual White Paper: https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9000/nb-06-cat-9k-stack-wp-cte-en.xhtml

Chapter 23. Infrastructure as Code (IaC)

In this chapter, you will learn about the following:

- Scaling up network deployment via automation
- Working with structured data
- Using revision controls
- Building a data model
- Deploying an “as code” architecture
- Pre-validation in the physical replica or a digital twin

Introduction

For the past few decades, network infrastructures have been configured in relatively consistent ways. The deployment, provisioning, and maintenance of devices take place via a command-line interface using applications such as SSH (or Telnet), using the web-based graphical user interface on supporting platforms, or under some circumstances, provisioning using network management applications through protocols such as SNMP. While this approach has served us well over the decades, it has a number of inherent challenges associated with it. While there are newer capabilities and methods such as NETCONF, RESTCONF, and model-driven telemetry, many organizations still lean heavily toward the use of legacy methods.

A new greenfield network deployment that is correctly designed is very much like a new car. At the beginning with a new vehicle, everything works. It drives safely and securely and has the latest software updates, allowing features like GPS maps to be up-to-date for navigation, and the latest driver safety and security features are in place. With time, however, things become outdated and potentially need to be updated, swapped out, or replaced entirely. In a network, over time, similar challenges present themselves. Network protocols that were deployed and in use in older versions may not have supported password authentication to exchange updates; security ciphers that were used for authentication protocols, such as SSH or DTLS for tunnel establishment, may not be up-to-date or may be proven to be insecure; software versions may not be the latest and most preferred versions; and in the worst cases, they may not even boot up given the "end of life" dates associated with the hardware in use.

In addition to software and security enhancements, the manual configuration of networks tends to be quite individualistic, where each network operator adds their own personal preference and creativity to apply a solution or solve a problem. For instance, to limit the distribution of routing information shared in the network protocol BGP, one operator may choose to select a route-map linked to an IP access-list to limit route propagation, whereas another user may choose to use a prefix-list. In other domains and areas such as network access control (NAC), one user may choose to leverage a dynamic ACL (DACL) to secure a port, whereas another operator may choose a fixed ACL on the interface or an ACL applied in conjunction with an interface template.

While it is great to have the flexibility and variety of options to choose from when designing an architecture, it is that flexibility that can lead to a lack of cohesiveness and standards within a network architecture, resulting in the network accumulating baggage, including configuration snowflakes. What is meant by this is that the network, over time, without the right resilience structures, checks, and adherence to golden standards, can eventually end up in a state that is operationally difficult to manage, difficult to automate, and in the worst cases, insecure.

This chapter aims at providing an overview of Infrastructure as Code (IaC) in the context of security and resilience considerations that should be taken into account in approaching a programmability-based deployment. For a deeper dive into network automation and model-driven network deployment, *Network Automation Made Easy* (Cisco Press) and *Model-*

Driven DevOps (Addison-Wesley Professional) provide a great deep dive into the automation aspects associated with this journey.

Evolution of Automation in Network Device Deployment and Management

In many domains, the size of the networks began to make manually deploying them impossible to achieve; this challenge was most apparent in service provider networks but also became commonplace in industrial and enterprise environments, where millions of end users would connect and leverage the networks. To deal with the increase in demand and the run rate of network deployments, organizations commonly began to use scripts to deploy networks. Initial scripting languages used for these functions were simplistic. The use of tools such as Perl and Expect, as shown in Example 23-1, started to become commonplace.

Example 23-1 *Using Expect to Interact with Network Devices*

```
#!/usr/bin/expect

set timeout 60
log_file tsec.log
spawn ssh admin@10.20.30.40
expect "Password:" { send "1112!\r" }
expect "#" { send "terminal length 0\r" }
expect "#" { send "show cts environment\r" }
expect "#" { send "exit\r" }

./expect
spawn ssh admin@10.20.30.40
(sdaadmin@10.20.30.40) Password:

Branch-FIAB#terminal length 0
Branch-FIAB#show cts environment
CTS Environment Data
```

```
====================
Current state = COMPLETE
Last status = Successful
Service Info Table:
Local Device SGT:
  SGT tag = 2-09:TrustSec_Devices
Server List Info:
Installed list: CTSServerList1-0001, 1 server(s):
 *Server: 192.168.70.101, port 1812, A-ID 2981E092AD240599BB43C12
          Status = ALIVE
          auto-test = TRUE, keywrap-enable = FALSE, idle-time = 6
Security Group Name Table:
    0-02:Unknown
    2-09:TrustSec_Devices
    3-00:Network_Services
    4-36:Employees
```

In networks, these tools did help in improving some efficiency in terms of creating configuration changes and taking a more centralized approach to the configuration of systems. This approach, while much better than the manual configuration of systems, tended to be relatively simplistic: they might have one script per job and often one single owner per script who was responsible for its maintenance, with limited collaboration between teams in the context of usage and standards. Consequently, the overall impact achieved through its usage would tend to be somewhat limited. Early Perl or Expect scripts often were not written in a multithreaded manner, with execution being serial and performed device by device rather than in parallel, leading to long execution times for tasks across a network.

With the trend of operators shifting toward Python instead of Perl and Expect, new modules and frameworks enabled operators to execute scripts more easily in more simplified ways. While the scripting languages that were used improved the ability to write simple, meaningful, and scalable code constructs, interfacing with systems through a Secure Shell CLI (which was created for human rather than machine-to-machine interaction) retained the inherent problem of being useful operationally beyond the execution of simple configuration tasks. The reason for this was that outputs derived from systems needed to be post-parsed with regular expressions to derive meaning (often referred to as *screen scraping*). This task, while achievable, can be error prone due to potential changes in the CLI outputs derived from release to release, which in turn could impact the script’s ability to derive needed data.

Due to the inherent challenges in networking, with centralized command and control systems only being able to provide a limited network view from their vantage point, as a result of equal cost network paths being used, that can result in network health checks only checking a single path through the network.

The path that may be in use from the central system will be limited in terms of its view of network latency, and applied security rules and logic, which may be applied at selective points in the network that may not apply to the respective path in use. Due to these limitations, distributed script execution became an attractive prospect for certain activities. The introduction of the TCL scripting language into Cisco routers and switches opened the door to such execution being possible.

Example 23-2, based on an initial prototype by Jónatan Þór Jónasson, is a script that is called remotely but can be executed via TCL locally on a switch.

Example 23-2 *TCL Script Example That Can Be Remotely Launched from an IOS CLI*

```
if {$argc != 3} {
       puts "Please ensure that all variables are included"
       puts "Syntax:"
       puts "tftp://<ip_address>/WakeUp.tcl Broadcast_Address Tar
   } else {
       set broadcastAddr [lindex $argv 0]
       set macAddr [lindex $argv 1]
       set vrfTable [lindex $argv 2]
       puts "\n\n======== Wake on Lan Script ========\n\n"
       puts "Selected Broadcast Address: $broadcastAddr"
       puts "Selected Mac Address: $macAddr"
       puts "Selected VRF Table: $vrfTable"
   }

# Function to parse mac address for transmission
proc WakeOnLan {broadcastAddr macAddr vrfTable} {
     set net [binary format H* [join [split $macAddr -:] ""]]
     set pkt [binary format c* {0xff 0xff 0xff 0xff 0xff 0xff}]

     for {set i 0} {$i < 16} {incr i} {
        append pkt $net
     }

     # Open UDP Socket and Transmit the Wake on Lan Magic Packet.
     set udpSock [udp_open]
     fconfigure $udpSock -translation binary \
          -broadcast 1 \
          -remote [list $broadcastAddr 4580] \
          -vrf $vrfTable
     puts $udpSock $pkt
     flush $udpSock;
     close $udpSock
}
set i 0
while {$i < 10} {
    incr i
    WakeOnLan $broadcastAddr $macAddr $vrfTable
    puts  -nonewline "!"
}
```

As time progressed, networking platforms moved beyond the limitations of the TCL scripting language and started to introduce more advanced ways to support interaction. They included NETCONF/Yang and RESTCONF capabilities, which were introduced into newer Cisco software versions of network operating systems (IOS XR, IOS XE, NXOS). The advantage to using these new methods is the ability to more easily derive value through structured data, as shown from an IOS XE device in Example 23-3.

Example 23-3 *RESTCONF Output*

```
Edge#show run int gig 1/0/1 | format restconf-json
{
  "data": {
    "Cisco-IOS-XE-native:native": {
      "interface": {
        "GigabitEthernet": [
          {
            "name": "1/0/1",
            "description": "Fabric Physical Link",
            "switchport-conf": {
              "switchport": false
            },
            "isis": {
              "Cisco-IOS-XE-isis:network": {
                "point-to-point": true
              }
            },
            "clns": {
              "Cisco-IOS-XE-isis:mtu": "1400"
            },
            "bfd": {
              "Cisco-IOS-XE-bfd:interval-interface": {
                "msecs": 250,
                "min_rx": 250,
                "multiplier": 3
              }
            },
            "dampening": {
            },
            "ip": {
              "address": {
                "primary": {
                  "address": "172.16.10.69",
                  "mask": "255.255.255.254"
                }
              },
              "pim": {
                "Cisco-IOS-XE-multicast:pim-mode-choice-cfg": {
                  "sparse-mode": {
                  }
                }
              },
              "router": {
                "Cisco-IOS-XE-isis:isis": {
                }
              },
              "redirects": false
            },
            "load-interval": 30,
            "Cisco-IOS-XE-cts:cts": {
              "role-based": {
                "enforcement-switching": false
              }
            }
          }
        ]
      }
    }
```

```
    }
}
```

In Cisco platform software in IOS XE, IOS XR, and NXOS, when using NETCONF, you are able to use the commit confirmation functionality. This specific capability allows a change to be applied and automatically rolled back if a follow-up confirmation is not sent by the operator within a given period of time. To execute this capability, you use candidate data stores ; this provides a further level of control in applying network changes to avoid disruption and isolation of network nodes.

When working with candidate data, you can lock configuration changes, which enables you to avoid issues arising through concurrent changes taking place. Figure 23-1 shows the steps involved in locking the configuration during planned configuration changes.

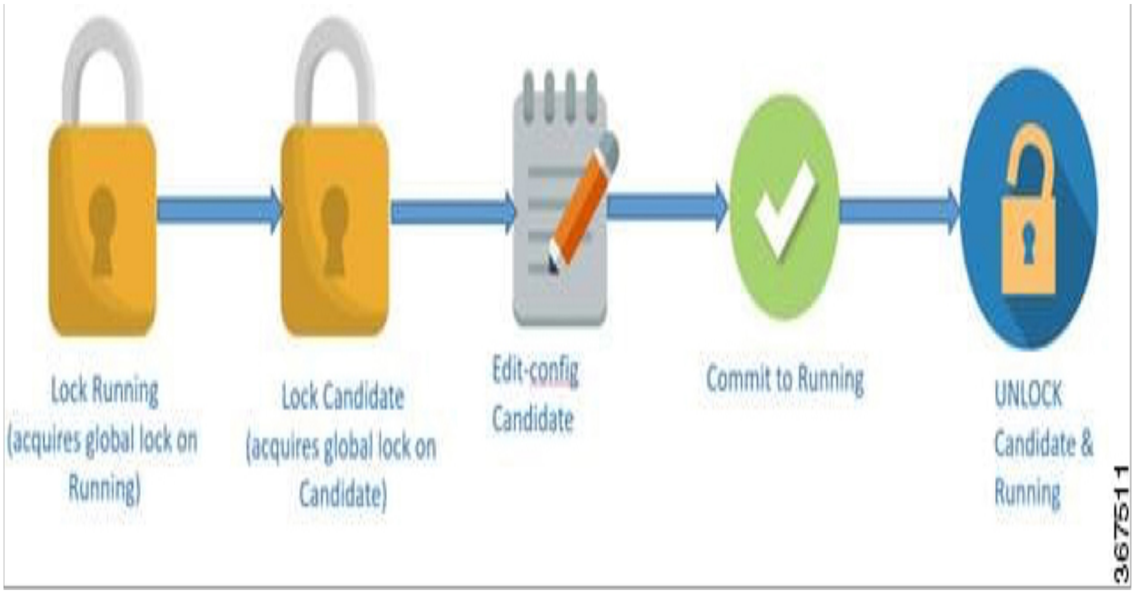

Lock Running (acquires global lock on Running)
Lock Candidate (acquires global lock on Candidate)
Edit-config Candidate
Commit to Running
UNLOCK Candidate & Running
367511

Figure 23-1 *Locking and Unlocking Candidate and Running Configurations During Provisioning*

Example 23-4 shows the programmability-based execution of a Python script that uses NETCONF to configure a system without a candidate datastore configured to perform a commit-confirmed execution. As you can see, the script shuts down the one and only interface that the node has, resulting in the script losing its connection to the router.

Example 23-4 *Service Impacting Change Without Commit Confirmed and Candidate Data Store*

```
python3 netconf_basic.py

Verifying the reachability to network node.

Netconf capabilities on configured node:

urn:ietf:params:netconf:capability:writable-running:1.0
urn:ietf:params:netconf:capability:rollback-on-error:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:netconf:capability:notification:1.0
urn:ietf:params:netconf:capability:interleave:1.0
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=e
tagged,report-all
urn:ietf:params:netconf:capability:with-operational-defaults:1.0?
all-tagged,report-all
urn:ietf:params:netconf:capability:yang-library:1.0?revision=2019
id=87cabe09133b813c1a59c6f41e742fcd
urn:ietf:params:netconf:capability:yang-library:1.1?revision=2019
id=87cabe09133b813c1a59c6f41e742fcd

NETCONF configuration to be pushed:

<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <interface>
            <GigabitEthernet>
                <name>1</name>
                <shutdown/>
            </GigabitEthernet>
        </interface>
    </native>
</config>

Failed to shut down GigabitEthernet1 interface or lost reachabili
waiting for an rpc reply.

Network node no longer reachable via ICMP
```

Example 23-5 (https://github.com/joshhalley/Zero_Trust_in_Resilient_Cloud_and_Network_Architectures/Chapter_23/netconf_advanced.py) shows a candidate configuration datastore configured in conjunction with a rollback timer. In this example, unlike the first example (Figure 23-4), the respective configuration rolls back, avoiding an extended loss of service.

Example 23-5 *Service Impacting Changes Using Candidate Data Store with Confirmed Commit*

```
python3 netconf_advanced.py (other)

Verifying the reachability to network node.

Listing existing netconf capabilities on configured node:

urn:ietf:params:netconf:capability:confirmed-commit:1.1
urn:ietf:params:netconf:capability:confirmed-commit:1.0
urn:ietf:params:netconf:capability:candidate:1.0
urn:ietf:params:netconf:capability:rollback-on-error:1.0
urn:ietf:params:netconf:capability:validate:1.0
urn:ietf:params:netconf:capability:validate:1.1
urn:ietf:params:netconf:capability:xpath:1.0
urn:ietf:params:netconf:capability:notification:1.0
urn:ietf:params:netconf:capability:interleave:1.0
```

```
urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=e
tagged,report-all
urn:ietf:params:netconf:capability:with-operational-defaults:1.0?
all-tagged,report-all
urn:ietf:params:netconf:capability:yang-library:1.0?revision=2019
id=cc85f3e8a79595066bd141b05158785b
urn:ietf:params:netconf:capability:yang-library:1.1?revision=2019
id=cc85f3e8a79595066bd141b05158785b
NETCONF configuration to be pushed:
NETCONF configuration to be pushed:

<UPDATED SCRIPT NEEDED - used other.py>
```

Looking at another example shown in Figure 23-2, one of our customers had the challenge of maintaining a view of routers that they had deployed in the field with their customer base. Given that their customers would often move devices to different public handoff points, keeping track of the active devices was becoming cumbersome. To resolve this challenge, the customer decided to deploy a push/pull construct through a combination of IP SLA and RESTCONF communications.

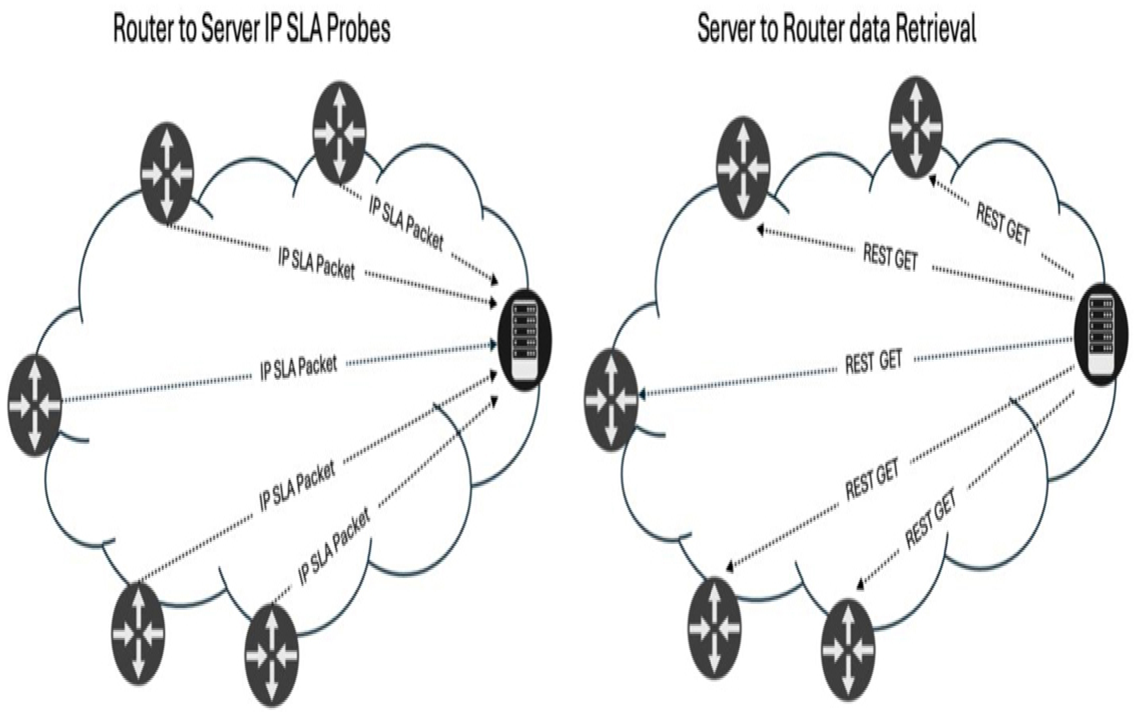


Figure 23-2 *Topology for IP SLA RESTCONF Data Retrieval*

The routers in the field are configured with an IP SLA probe to interface with a public device on a fixed TCP socket, sending a known string, as shown in Example 23-6.

Example 23-6 *Router Configuration, Triggering Call Home Events for a Remote Script via IP SLA*

```
IOS-XE (Configuration):

! username for restconf access
username restconf-user privilege 15 secret 0 Cisco123

! limit restconf access to source network 10.0.0.0/8 only (option
ip access-list extended RESTCONF-ALLOWED
 10 permit tcp 10.0.0.0 0.255.255.255 any

restconf
restconf ipv4 access-list name RESTCONF-ALLOWED (optional to limi
ip http authentication local
ip http secure-server
ip http secure-port 30413 (RESTCONF ports we expect connections t
ip http secure-trustpoint <trustpoint-name>

ip sla 1
 tcp-connect <server-ip> 60000 source-port 60000 control disable
to server-ip from source port 60000 to destination port 6000) eve
  tos 184 (DSCP EF to mark traffic, optional - ISP might rewrite
  frequency 300
ip sla schedule 1 life forever start-time now
```

On the server side, as shown in Example 23-7, a configured daemon is set up to cache the probe request and attempt to build a RESTCONF connection, retrieving relevant information about the node.

Example 23-7 *Server-Side Script Acting as a Daemon Responding to IP SLA*

```
#!/usr/bin/python3
import requests
requests.packages.urllib3.disable_warnings()
import json
import socket
import datetime

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(("0.0.0.0", 60000))

s.listen()

while True:
    # Wait for IP SLA tcp-connect
    connection, remote_peer = s.accept()

    # Close connection, return code on router for this SLA will b
    connection.close()

    # Get timestamp
    local_dt = datetime.datetime.now().strftime("%a %d.%m.%Y %H:%

    # Remote SRC port needs to be 60000
    if not remote_peer[1] == 60000:
        print(f"{local_dt} - Connection does not come from remote
{remote_peer[0]}:{remote_peer[1]}")
        continue

    # Remote SRC port is TCP 60000 (IP SLA tcp-connect from route
    print(f"{local_dt} - TCP Connection from: {remote_peer[0]}:{r

    # Get timestamp
    local_dt = datetime.datetime.now().strftime("%a %d.%m.%Y %H:%

    # Send RESTCONF get request inventory of hw-type-chassis
    print(f"{local_dt} - Sending RESTCONF GET request to: {remote_
    headers = {"Accept": "application/yang-data+json"}

    try:
        response = requests.get(f"https://{remote_peer[0]}:30413/
oper:device-hardware-data/device-hardware/device-inventory=hw-typ
"Cisco123"), headers=headers, verify=False, timeout=5)
        # Will raise exception as well for HTTP error (return cod
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        print("Requests Exception:", e)
        continue
    # For our get request the expected response status code needs
continue and wait for the next request)
    if not (response.status_code == 200):
        print(f"Unexpected response status code: {response.status_
        continue

    # GET request was successful
    print(json.dumps(response.json(), indent=4))
    print()

s.close()
```

The resulting communication results in a JSON-formatted payload detailing specific attributes that are of interest for the network operator (shown in Example 23-8). Similar queries could easily be constructed for something that is more operationally significant for other network estates.

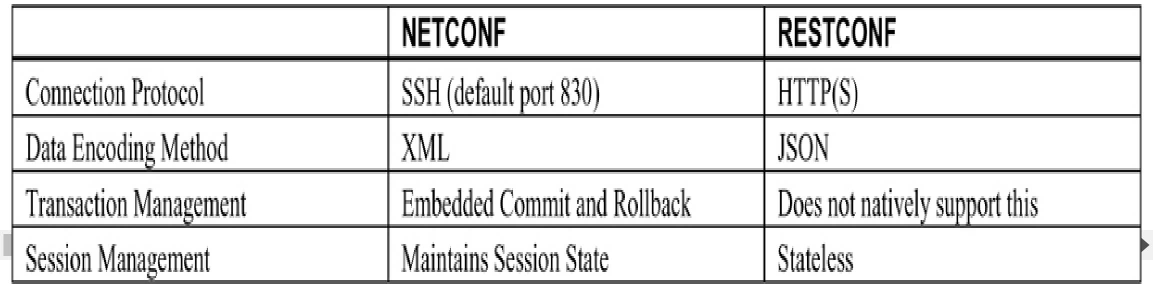
Example 23-8 *Outputs of Running Daemon upon Interaction*

```
Sat 14.09.2024 12:01:07 - TCP Connection from: 10.20.30.40:60000
Sat 14.09.2024 12:01:07 - Sending RESTCONF GET request to: 10.20.
{
    "Cisco-IOS-XE-device-hardware-oper:device-inventory": [
        {
            "hw-type": "hw-type-chassis",
            "hw-dev-index": 0,
            "version": "V01",
            "part-number": "C1111-NBJERY3D",
            "serial-number": "ZAP34948111",
            "hw-description": "Cisco C1111-NBJERY3D Chassis",
            "dev-name": "Chassis",
            "field-replaceable": true,
            "hw-class": "hw-class-physical"
            }
        ]
}
```

While similar actions could be performed over protocols like SSH, RESTCONF provides a more structured data payload that is more simplistic to parse and use in network automation functions and/or data processing.

Through these formalized and reliable methods of interacting with data and performing changes, many organizations find that a programmability-first approach becomes more simplistic to manage and to work with. Table 23-1 lists some of the differences between NETCONF and RESTCONF.

Table 23.1 *Differences Between NETCONF and RESTCONF*



	NETCONF	RESTCONF
Connection Protocol	SSH (default port 830)	HTTP(S)
Data Encoding Method	XML	JSON
Transaction Management	Embedded Commit and Rollback	Does not natively support this
Session Management	Maintains Session State	Stateless

Examples 23-1, 23-4, and 23-5 showcase the interaction with infrastructure systems achieved via programmability-based methods. What these examples do not immediately achieve is automation. It would be incorrect to simply consider that the use of a script or programmability method is truly advanced automation; it does perform a task in a more rapid and automated manner than a human operator typing out each and every command. If the script is to be executed multiple times manually for its respective execution, however, it cannot truly be defined as advanced automated. In site-reliability engineering terms, the repeated human execution of a script is often referred to as *toil*.

So how can an organization get from manually executing monolithic scripts that perform a function or task toward what would be considered true automation? This is a question that should be asked based on the business needs and goals and identified through careful introspection of an end-to-end process flow and logic.

Let's consider that an organization is responsible for manufacturing widgets. This organization has 25 different tasks that take place from the beginning of the fabrication of the widgets to their eventual delivery to customers. Identifying the systems that exist in the chain of events from start to end, where a human is in the loop and performs a task, can help in understanding whether the task that is being performed is unique and requires special innovation, creation, and efforts that vary each and every time, or whether the tasks being executed are monotonous or repetitive and take place every day or even multiple times per day. Looking at such a process can help a company understand areas for potential optimizations, such as linking backend ordering systems together with a logistics management system to ensure that the tracking of deliveries is properly handled or that deliveries that may take the same route can be performed within the same day. The same goes for potential issues that arise with the manufacturing process, tickets being created for issues, and their respective remediation and closure. You may be asking, "What does any of this have to do with IT networks?" and rightly so.

The reason we shared this example is that while most networks may not be associated with a widget production line (although some may be), the goal of optimizing processes and automating activities can be just as fruitful and yield significant improvements in the way that operations take place. Such optimizations can be key in reducing costs in a business and identifying the correct trends and areas to focus to perform adjustments and changes. The application of *process mining*, while outside the scope of this book, can provide powerful techniques for using data from information systems to analyze business processes and identify areas for optimizations and improvements.

In the context of IP-based networks, efficiency gains can generally be achieved in various areas, from shifting the provisioning and operations of a network to an Infrastructure as Code method, as we will outline further in this chapter, and tasks such as linking the ticketing system (ITSM) into the automated fault detection and network monitoring infrastructure—the dispatch of tickets and updates of remediation being automatically shared with users via collaboration tools such as Webex and approval or change requests via DUO.

Note that automation should not be limited to the IP network estate of routers and switches but should encompass the broader IT infrastructure that exists, including firewalls, third-party systems, and even dependent systems that may be related to the core business of the organization with which you work to achieve the maximum value possible.

Working with Structured Data

Moving away from using a CLI to using a structured data and programmability-based approach to create meaningful configuration in scripting languages or using tools like Postman can represent a new learning curve for some. As daunting as it may seem at first glance, the change can quickly start showing rewards, through a more expedient means of parsing and deriving meaning from data.

Let's look at two different outputs, Example 23-9 and Example 23-10, showing standard Cisco CLI output for an interface configuration. This represents a set of unstructured data; a user who is attempting to parse and search for something specific like a regular expression would likely need to use this output to derive the sought-after value.

Example 23-9 *Unstructured Cisco CLI Output*

```
Edge#show run int gig 1/0/1
Building configuration...

Current configuration : 329 bytes
!
interface GigabitEthernet1/0/1
 description Fabric Physical Link
 no switchport
 dampening
 ip address 172.16.10.70 255.255.255.254
 no ip redirects
 ip pim sparse-mode
 ip router isis
 load-interval 30
 no cts role-based enforcement
 bfd interval 250 min_rx 250 multiplier 3
 clns mtu 1400
 isis network point-to-point
end
```

Example 23-10 *Regular Expression in Python Script (Using the RE Module)*

```
<snip>
pattern = r"ip address (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}) (\d{1

# Search for the pattern in the output
match = re.search(pattern, output)

# Check if the pattern is found
if match:
    ip_address = match.group(1)
    subnet_mask = match.group(2)
    print(f"IP Address: {ip_address}")
    print(f"Subnet Mask: {subnet_mask}")
else:
    print("No IP Address and Subnet Mask found.")
```

Unlike Example 23-9, Example 23-11 deals with structured JSON data, which provides the benefit of fixed, known paths to key information such as IP addressing. In the RESTCONF example, retrieving the IP address details does not require a regular expression to identify a pattern; instead, it simply provides a known path for its respective retrieval.

Example 23-11 *Structured RESTCONF Output*

```
Edge#show run int gig 1/0/1 | format restconf-json
{
  "data": {
    "Cisco-IOS-XE-native:native": {
      "interface": {
        "GigabitEthernet": [
          {
            "name": "1/0/1",
            "description": "Fabric Physical Link",
            "switchport-conf": {
              "switchport": false
            },
            "isis": {
              "Cisco-IOS-XE-isis:network": {
                "point-to-point": true
              }
            },
            "clns": {
              "Cisco-IOS-XE-isis:mtu": "1400"
            },
<SNIP - removed for brevity>
            "ip": {
              "address": {
                "primary": {
                  "address": "172.16.10.69",
                  "mask": "255.255.255.254"
                }
              },
              "pim": {
                "Cisco-IOS-XE-multicast:pim-mode-choice-cfg": {
                  "sparse-mode": {
                  }
                }
              },
              "router": {
                "Cisco-IOS-XE-isis:isis": {
                }
              },
              "redirects": false
            },
            "load-interval": 30,
            "Cisco-IOS-XE-cts:cts": {
              "role-based": {
                "enforcement-switching": false
              }
            }
          }
        ]
      }
    }
  }
}



Python Script to print IP Addresses:

import json

# JSON-like output provided as a string for demonstration purpose
output = '''
{
    "data": {
        <SNIP removed for brevity>
}
'''
```

```
# Parse the JSON output
data = json.loads(output)

# Extract the interface data
interfaces = data['data']['Cisco-IOS-XE-native:native']['interfac

# Loop through the interfaces to find and print the IP address an
for interface in interfaces:
    ip_info = interface.get('ip', {}).get('address', {}).get('pri
    ip_address = ip_info.get('address', 'No IP address found')
    subnet_mask = ip_info.get('mask', 'No subnet mask found')
    print(f"Interface {interface['name']}: IP Address: {ip_addres
```

In Examples 23-11 and 23-3, you can see structured data in the form of NETCONF (with XML) and RESTCONF (with JSON). The advantages to making use of the RESTCONF/NETCONF methods when performing changes in the network is highly beneficial, due largely to the ability to leverage syntax verification and commit confirmation. Although commit confirmation has long existed in IOS XR software, which is commonly used in service providers, it has not been possible (at least not over the CLI) in other Cisco network operating system software.

Revision Control

Maintaining an up-to-date view of the state of software and changes that are made when developing software has long been considered a best practice. Version control systems such as Git and Bitbucket have supported large software projects, allowing developers to work on the same code base in parallel and easily merge in updates, new features, changes, and bug fixes.

In the domain of network administration, approaches that involve advanced version control functionalities have been taken to a lesser extent for change management; however, these approaches are still seen and used today to support verifying old versus new configurations as part of post-mortem analysis of configuration changes that may have resulted in a service having an impact on the network.

Although Example 23-12 may be useful for an operator who is willing to connect system by system to troubleshoot, it does lack the capabilities that are awarded via a proper central system that can manage version control. Some of Cisco’s products, such as Catalyst Center, do provide various levels of configuration revision and, under certain circumstances, compliance validation.

Example 23-12 *Using a Configuration Archive in IOS XE to Verify Configuration Differences Locally*

```
IS_GW2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
IS_GW2(config)#int gig 3
IS_GW2(config-if)#description New Description
IS_GW2(config)#end
IS_GW2#wri mem
IS_GW2#show archive
The maximum archive configurations allowed is 10.
There are currently 5 archive configurations saved.
The next archive file will be named bootflash:/archive-<timestamp
 Archive #   Name
   1         bootflash:/archive-Sep-22-13-26-01.406-0
   2         bootflash:/archive-Sep-22-13-27-03.928-1
   3         bootflash:/archive-Sep-22-13-27-53.389-2
   4         bootflash:/archive-Sep-22-13-29-31.912-3
   5         bootflash:/archive-Sep-22-13-30-53.915-4 <- Most Rece
<SNIP>
IS_GW2#show archive config differences bootflash:/archive-Sep-22-
!Contextual Config Diffs:
<SNIP>
interface GigabitEthernet3
 -description New Description
```

```
IS_GW2#
```

When looking at something that is multivendor or multisystem compatible, many operations teams default back to the software world for their best practices and adopt Git or Bitbucket for their existing capabilities.

One example of how this can be achieved is shown in Example 23-13. After a configuration is saved, the file is written to a central server (using a more secure method than TFTP is strongly recommended).

Example 23-13 *Event Manager[nd]Triggered Configuration Backups*

```
event manager applet Config_Backup
 event cli pattern "(write|write memory|copy running-config start
 action 0.1 info type routername
 action 1.0 cli command "enable"
 action 1.1 cli command "copy run tftp" pattern "Address"
 action 1.2 cli command "10.90.90.7" pattern "filename"
 action 1.3 cli command "XYZCORP/INFRA/$_info_routername-config.t
 action 2.0 syslog priority informational msg "Configuration chan
executed"
```

Once the file is written to the central server, a further cronjob on the receiving server can run. This job commits new files that are seen in a configured Git repository. For an example of how to set this up, see the following steps and Example 23-14:

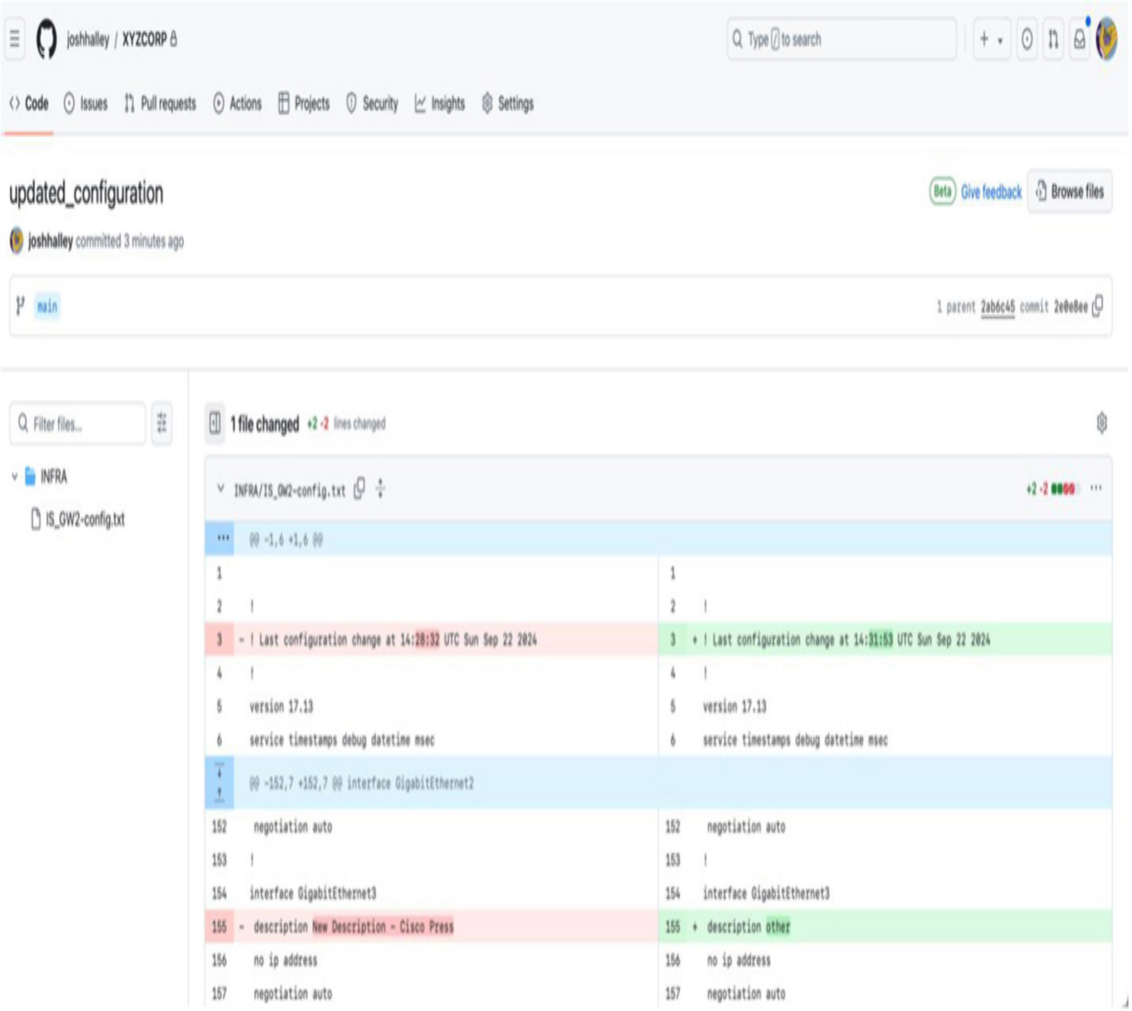
1. Create a repository in your Git server.
2. Clone the repo to the SCP/SFTP/FTP/TFTP server that you are writing your configurations to as follows:
   tftpserver:/var/lib/tftpboot$ git clone git@github.com:joshhalley/XYZCORP.git
3. Log in to your Git repo with your username.

4. Check that files uploaded on change can be committed into the repo.

Example 23-14 *Git Configuration for Archiving Changes*

```
tftpserver:/var/lib/tftpboot/XYZCORP/INFRA$ git commit -a -m "upd
[main 2e0e8ee] updated_configuration
 1 file changed, 2 insertions(+), 2 deletions(-)
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 393 bytes | 393.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local obje
To github.com:joshhalley/XYZCORP.git
   2ab6c45..2e0e8ee  main -> main
tftpserver:/var/lib/tftpboot/XYZCORP/INFRA$
```

5. Verify that revision control is working by comparing the adjusted configuration (see Figure 23-3).



updated_configuration

joshhalley committed 3 minutes ago

main 1 parent 2ab6c45 commit 2e0e8ee

1 file changed +2 -2 lines changed

INFRA/IS_GW2-config.txt +2 -2

```
@@ -1,6 +1,6 @@
  1                                                          1
  2   !                                                      2   !
  3 - ! Last configuration change at 14:28:32 UTC Sun Sep 22 2024    3 + ! Last configuration change at 14:31:53 UTC Sun Sep 22 2024
  4   !                                                      4   !
  5   version 17.13                                          5   version 17.13
  6   service timestamps debug datetime msec                 6   service timestamps debug datetime msec
@@ -152,7 +152,7 @@ interface GigabitEthernet2
152    negotiation auto                                    152    negotiation auto
153   !                                                    153   !
154   interface GigabitEthernet3                           154   interface GigabitEthernet3
155 -  description New Description - Cisco Press           155 +  description other
156    no ip address                                       156    no ip address
157    negotiation auto                                    157    negotiation auto
```

Figure 23-3 *Differences Seen in Github for Pushed Configurations*

Assuming everything is working as desired, you can update the crontab on the server's UNIX operating system to automatically commit changes for the directory where the files are written to. If the version control application you chose is Git, then no uploads will take place unless there is a change to the content of the files that are written. This would allow for an aggressive timer value (~1 minute) to be used within the configuration to ensure regular synchronization of the configurations.

Building a Data Model

When you are beginning to shift to an Infrastructure as Code (IaC) approach for architecture, creating a well-thought-out and structured data model is crucial to the success of this move. A data model consists of entity structures, relationships, and attributes to represent components that build up your respective “as code” architecture. Through the representation of the resources and relationships defined in the model, tools like Terraform can be used to provision and manage network resources declaratively. The generation of such a model should always be based on a clear objective, where all stakeholders are aligned with what the outcome should be and what needs to be achieved.

If we look at the broader context of applicability of data models, although some systems may have a level of overlap in their functionalities, fundamental differences can exist between platforms, software, and core functionality, and even in implementations for common features. In this case, creating a unified model doesn’t always simplify things. Creating models for platforms is often based on core and foundation needs initially—with model progression mapping updates and improvements based on customer requirements and new capabilities that may commit into newer software versions or hardware updates.

While model unification may not always make sense, using common semantic conventions for the development of data models that could be chained together may make usage and deployment more simplistic for the human operators and potentially AI agents that may be working with IaC solutions.

When it comes to determining what dependent structures should exist, how the relationships should look, and which attributes should be mapped within the model, such an exercise needs to be planned appropriately. This is also an exercise that should be performed by individuals who not only understand the concepts of automation but also understand the subject matter within the respective technology domain to be able to ensure that the right relationships are created and mandatory and critical attributes are placed within the right levels of the hierarchy to allow the model to scale appropriately.

In addition to model definition based on what exists today, it is also important to consider that technology does not tend to be static; it tends to evolve, grow, change, and improve with time. For this reason, modularity in the creation of the model is important. Considerations in planning data models should also anticipate what could be useful in the future, given the trajectory of a product. For instance, should the data model consider the potential future support for multitenancy although it is not part of the product today but is considered on the committed roadmap for the future? These sorts of decisions and trade-offs should be evaluated to avoid double work and overhead in the future.

When you're building out a data model, it is an optimal time to reflect on what should be tested for the respective attributes that are being included and which attributes are considered to be mandatory in terms of the data model's validity to be proven.

Figure 23-4 illustrates how such a data model can be generated, as depicted within an IaC implementation for Cisco ACI.

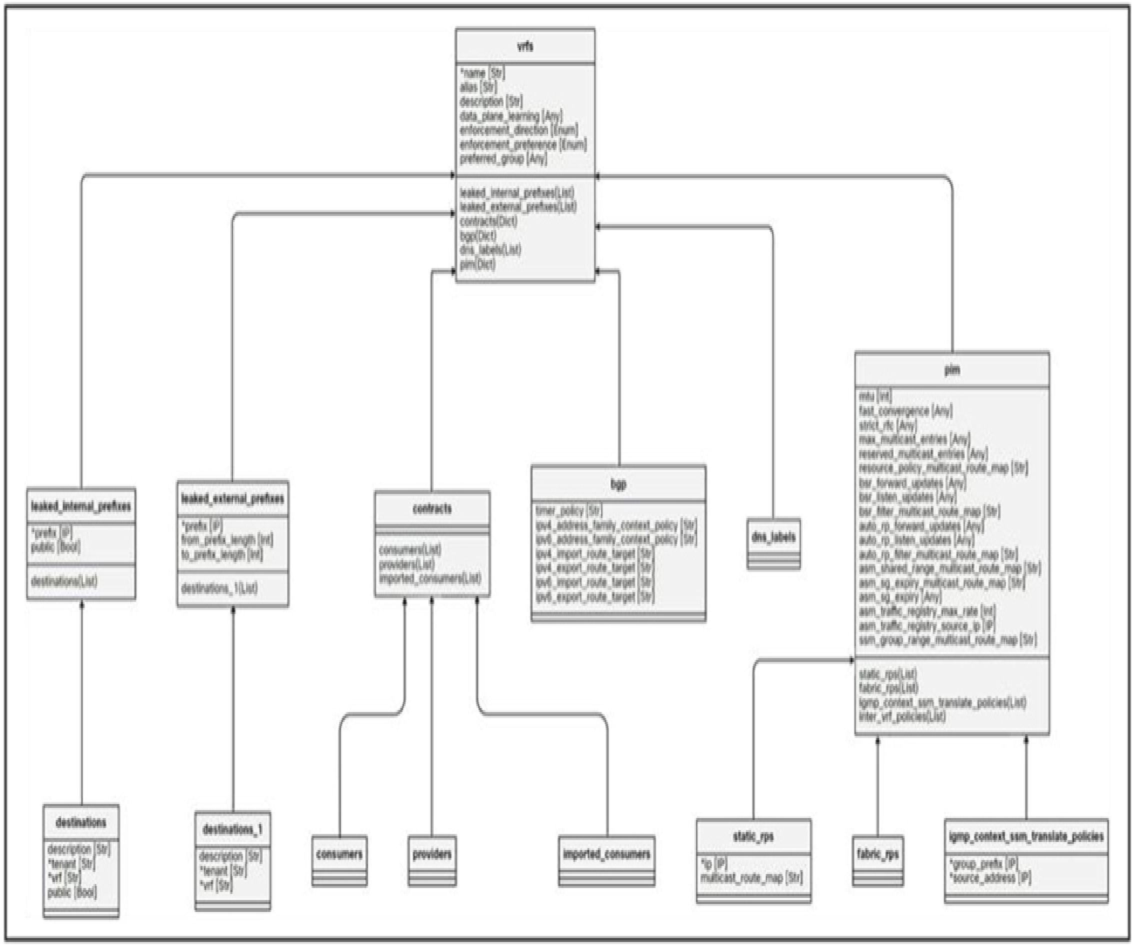

vrfs
*name [Str]
alias [Str]
description [Str]
data_plane_learning [Any]
enforcement_direction [Enum]
enforcement_preference [Enum]
preferred_group [Any]
leaked_internal_prefixes(List)
leaked_external_prefixes(List)
contracts(Dict)
bgp(Dict)
dns_labels(List)
pim(Dict)
leaked_internal_prefixes
*prefix [IP]
public [Bool]
destinations(List)
leaked_external_prefixes
*prefix [IP]
from_prefix_length [Int]
to_prefix_length [Int]
destinations_1(List)
contracts
consumers(List)
providers(List)
imported_consumers(List)
bgp
timer_policy [Str]
ipv4_address_family_context_policy [Str]
ipv6_address_family_context_policy [Str]
ipv4_import_route_target [Str]
ipv4_export_route_target [Str]
ipv6_import_route_target [Str]
ipv6_export_route_target [Str]
dns_labels
pim
mtu [Int]
fast_convergence [Any]
strict_rfc [Any]
max_multicast_entries [Any]
reserved_multicast_entries [Any]
resource_policy_multicast_route_map [Str]
bsr_forward_updates [Any]
bsr_listen_updates [Any]
bsr_filter_multicast_route_map [Str]
auto_rp_forward_updates [Any]
auto_rp_listen_updates [Any]
auto_rp_filter_multicast_route_map [Str]
asm_shared_range_multicast_route_map [Str]
asm_sg_expiry_multicast_route_map [Str]
asm_sg_expiry [Any]
asm_traffic_registry_max_rate [Int]
asm_traffic_registry_source_ip [IP]
ssm_group_range_multicast_route_map [Str]
static_rps(List)
fabric_rps(List)
igmp_context_ssm_translate_policies(List)
inter_vrf_policies(List)
destinations
description [Str]
*tenant [Str]
*vrf [Str]
public [Bool]
destinations_1
description [Str]
*tenant [Str]
*vrf [Str]
consumers
providers
imported_consumers
static_rps
*ip [IP]
multicast_route_map [Str]
fabric_rps
igmp_context_ssm_translate_policies
*group_prefix [IP]
*source_address [IP]

Figure 23-4 *Data Model Structure for ACI as Code*

As software versioning changes on platforms, features are added, and the data model evolves, it is important that version control systems are used in conjunction with the correct versioning of the data model. This allows the desired configuration definitions and the intent of the configuration to be stored in a version control system. In turn, this allows the model to grow with the platform and its respective capabilities, or with adjacent capabilities that may exist for “as code” deployments, which may involve using multiple related IaC projects together—for example, deploying “Meraki as Code” together with “ISE as Code.”

Network Controllers vs. Direct to Device

When you are deploying a network architecture using Infrastructure as Code, you may use various options to address the architecture. There are differing schools of thought on what the right and best approach is to maintain and operate a network architecture. The two different methods are to use a network controller[nd]based approach or a direct-to-device approach.

Advantages to using network controllers:

- Significant levels of vendor testing takes place for controllers and their corresponding automated configurations.
- Often, less overhead is required because a large-scale configuration template may already exist in a provisioning action from a controller API.
- Support from vendor technical support teams tends to be easier when dealing with an orchestrator solution that is heavily deployed.

Advantages to using direct-to-device:

- The exact configurations that are relevant for the given network can be deployed without extra overhead that is added from network orchestrators.
- Features that may exist on the platform prior to the orchestrator can be provisioned more rapidly.
- Advanced provisioning using syntax checks and confirmed commits can be applied on some network platforms.

- One less system is in the chain of execution when avoiding a network orchestrator.

Circling back to what the best approach is, the answer largely is "It depends." Both options have their merits. For a more standard approach, the orchestrator-based methods tend to be the most consumed by customers in the field today. That being said, we have also seen some very large customers have success in taking the direct-to-device approach when performing Infrastructure as Code operations.

Deploying an IaC Architecture

So far, we've touched on a number of key concepts that are related to the deployment of Infrastructure as Code, the use of data models, version control, and programmability-related tools. Although these components alone don't achieve very much, together they are the foundation for the deployment of an "as code[nd]based architecture."

In addition to the tools mentioned already, using further Infrastructure as Code tools is considered a best practice for such deployments:

- **Ansible:** This agentless YAML-based configuration engine executes a simplified list of tasks using a *playbook* to configure systems. Ansible has many plug-ins, which are called *collections* that can be added to simplify the tasks which are being performed. This allows for simplistic connectivity to systems through SSH to more advanced connectivity to systems through REST or other methods. Using Ansible is common in the system administrator domain, and its tools have been deployed within networking domains also. Some caveats with Ansible may become evident when attempting to deploy at a higher scale due to some of the older programming logic that was implemented in its base code.

- **Terraform:** Terraform, developed by HashiCorp, makes use of the HashiCorp Configuration Language (HCL), which is used for the deployment of "as code" architectures for systems that use their supported modules, which are referred to as *providers*. These modules allow specific system conventions to be used to map to the respective APIs utilized in their products. Terraform initially gained a lot of traction for use with hyperscalers, allowing a simple means to maintain a level of state for not only the provisioning of the architecture but also the further operation of it. Terraform is considered a popular, well-maintained, and reliable Infrastructure as Code provisioning ecosystem to use. The change in the licensing model from open source to business source licensing in August 2023 has, however, led to some concern in the industry around potential future costs that could arise from continued usage.

- **Open Tofu:** In response to the change of licensing within HashiCorp's Terraform, the open-source community, together with support from the Cloud Native Computing Foundation (a Linux Foundation project), forked the earlier open-source version of the Terraform project to remain a viable open-source offering for the community. At the time of this writing, OpenTofu 1.6.x was similar to Terraform 1.6.x, but with the variation in licensing, it is expected that the features that will commit into each project will likely diverge in the future.
- **Pulumi:** One further offering that is available for use for Infrastructure as Code projects is Pulumi, which is also an open-source offering. One of the key differentiators with Pulumi is that you can select a broad range of languages for use rather than needing to use HCL. This means that developers who are familiar with Go, Python, Java, TypeScript, and markup languages like YAML and CUE do not need to adjust to the "as code'" tools but rather can continue to work in the languages that they are familiar with.

Once the "as code" tools have been selected for the project, and the correct data model is in place, the provisioning of the architecture can commence. Because Infrastructure as Code is based around the principles of DevOps, this means that the correct developer frameworks and architectures are expected to be present to properly deploy as code configurations.

The DevOps approach provides the execution gates to ensure that architectures which are administered using code follow a similar approach to software development, ensuring that configurations are being maintained in a version control system and that configuration changes and updates are applied through rigorous validation, plus pre- and post-testing to lower the possibility of issues and errors, thus leading to more structured, robust, and successful deployments. The infinity racetrack shown in Figure 23-5 is often depicted to outline the key principles around continuous testing and continuous deployment. In the software world, this approach has reduced the time to deploy code—from releases being pushed out weekly or even monthly to new releases being generated nightly. In the broader context of Infrastructure as Code, two CI/CD pipelines should be considered: first, the pipeline for the development of the automation software and the development of the terraform modules and validation checks associated with this development exercise, and second, the lifecycle of the configuration changes and state in the network.

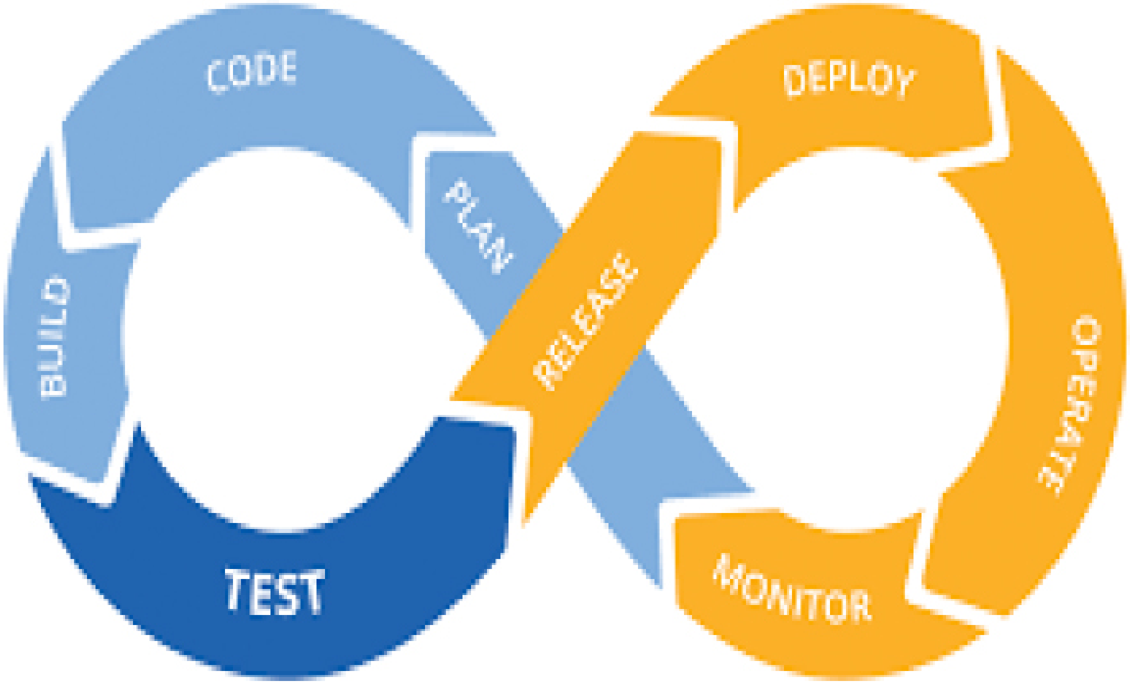


Figure 23-5 *The Infinity Racetrack*

When we look at how this approach juxtaposes itself with the infrastructure domain, it is quite simple. Let’s consider that an end-to-end network solution consists of firewalls, AAA servers, on-premises DCs, cloud-based DCs, along with LAN, WAN, and WLAN components, all of which need to work harmoniously with one another. Often disparate teams administer and test these components, performing their testing independently from one another. By taking a DevOps approach with continuous testing and continuous delivery, each team’s changes can be validated in its own platform-dependent (PD) pipeline, with a final complete end-to-end build, only undergoing platform-independent (PI) testing upon the successful completion of the PD pipeline execution. A consequence of this approach is that there could be a software update or configuration change that causes an issue in the AAA domain. If this pipeline failed, it would not be committed into the new nightly deployment sequence. However, other pipelines for changes that were successful could go ahead without the failed dependency being integrated, instead using an earlier build/configuration/setup that was proven to be functional and working. An example of how such a cascading CI/CD pipeline setup can look is shown in Figure 23-6.

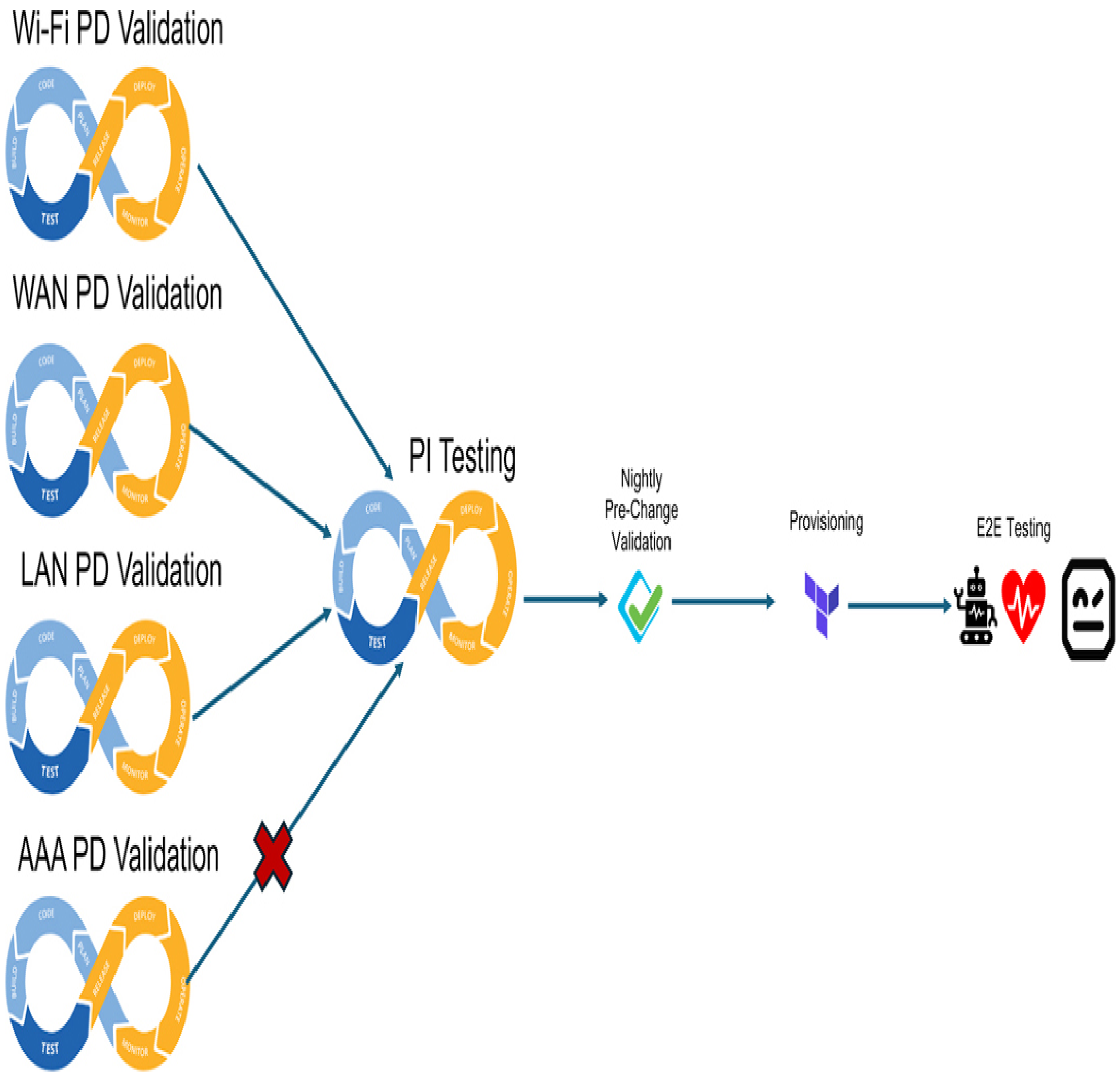


Figure 23-6 *Cascading CI/CD Pipelines*

Shifting how the network is managed to make use of advanced automation can bring significant value to the efficiency and stability in the operating IT infrastructure. However, for many, this is a big change, and without the right backing from leadership, it is difficult to execute. As networks grow in complexity and the security and span of control continue to become more difficult to manage, taking an automated approach is a good way to ensure that the right controls, safeguards, checks, and standards are in place to ensure the estate is as robust and secure as possible. To achieve this goal, it is essential that pre-validation and rigorous testing take place both pre- and post-deployment within "as code" networks to ensure that the right checks and balances are maintained.

Securing IaC Provisioning

As with any legacy means and methods to deploy configurations on critical infrastructure, maintaining the right security protocols is paramount. Legacy network protocols such as SNMPv1, v2c, and Telnet were superseded by more secure protocols such as SNMPv3 and SSH, allowing for more complex use of encryption methods and ciphers. In addition to utilizing secure transport methods for control-based configurations, system-level configurations represent a key aspect of security and hygiene on network nodes, including device administration rule sets, such as authorization rule mappings, role-based access control, and access control lists to limit systems that can communicate with the devices.

NETCONF Service-Level Restrictions:

You can limit access to NETCONF capabilities on supporting routers and switches in a number of ways. For example, you can block system access to NETCONF capabilities per IP address or use more elaborate role-based access limitations.

The most rudimentary level of device administration control that exists is session-level NETCONF session restriction, which can be configured as shown in Example 23-15.

Example 23-15 *Session-Level NETCONF Session Restriction*

```
Device# enable
Device# configure terminal
Device(config)# ip access-list standard NETCONF_HOSTS
Device(config-std-nacl)# permit 10.90.90.0 0.0.0.255
Device(config-std-nacl)# deny any
Device(config-std-nacl)# exit
Device(config)# netconf-yang ssh ipv4 access-list name NETCONF_HO
Device(config)# end
```

Once this restriction is configured, attempts to connect to the system via NETCONF for provisioning are restricted, resulting in negotiation failures occurring on systems attempting to connect to the nodes:

```
ncclient.transport.errors.SSHError: Negotiation failed: Error read
Connection reset by peer
```

You can use a similar command syntax when configuring hardening rules for RESTCONF-based system access, as shown in Example 23-16.

Example 23-16 *RESTCONF Service-Level Restrictions*

```
Device# enable
Device# configure terminal
Device(config)# ipv6 access-list RESTCONF_HOSTSv6
Device(config-ipv6-acl)# permit ipv6 2001:db8::1/32 any
Device(config-ipv6-acl)# deny ipv6 any any
Device(config-ipv6-acl)# exit
Device(config)# restconf ipv6 access-list name RESTCONF_HOSTSv6
Device(config)# end
```

You can achieve further device hardening in the context of the SSH server itself through the base configuration on the CLI in IOS XE. Example 23-17 provides an example of the default ciphers and attributes that are present during a day zero installation.

Example 23-17 *NETCONF Algorithm and Cipher Overview*

```
IS_GW2#show netconf-yang ssh server
Algorithm                        Type          Status
-------------------------------------------------------
rsa-sha2-256                     Hostkey       Enabled
rsa-sha2-512                     Hostkey       Enabled
ssh-rsa                          Hostkey       Enabled
aes128-ctr                       Cipher        Enabled
aes192-ctr                       Cipher        Enabled
aes256-ctr                       Cipher        Enabled
aes128-cbc                       Cipher        Enabled
aes256-cbc                       Cipher        Enabled
hmac-sha2-256                    MAC           Enabled
hmac-sha2-512                    MAC           Enabled
hmac-sha1                        MAC           Enabled
diffie-hellman-group14-sha1      KEX           Enabled
diffie-hellman-group14-sha256    KEX           Enabled
diffie-hellman-group16-sha512    KEX           Enabled
ecdh-sha2-nistp256               KEX           Enabled
ecdh-sha2-nistp384               KEX           Enabled
ecdh-sha2-nistp521               KEX           Enabled
```

Looking beyond limiting network access to a specific range for device administration, further security capabilities exist through the use of the NETCONF Access Control Module (NACM), which was standardized in RFC 6536. It can be used in conjunction with model-based AAA on supporting platforms.

The advantage of NACM is that a rich set of policy-based rules can be constructed in NETCONF and be used in conjunction with the AAA architecture.

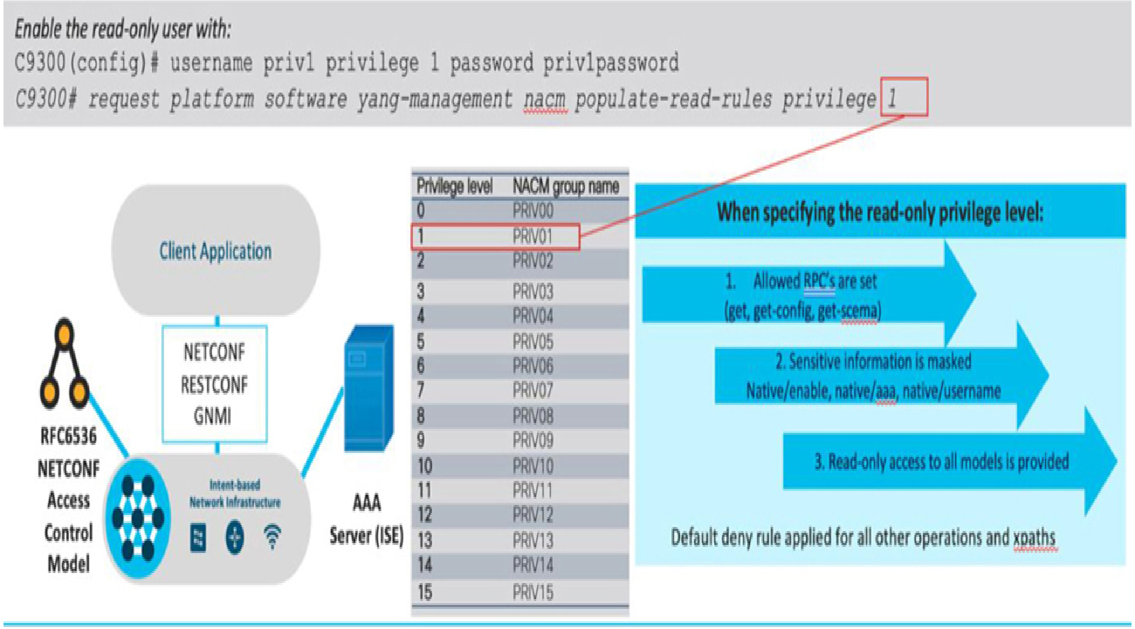
Example 23-18 represents the default configuration for NACM with IOS XE.

Example 23-18 *Verification of NACM Policy Applied to a Cisco IOS XE Device*

```
python3 verify_nacm.py
<?xml version="1.0" ?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:
message-id="urn:uuid:e2ff34f9-e236-4242-bc44-2e1abc5c9206">
  <data>
    <nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
      <enable-nacm>true</enable-nacm>
      <read-default>deny</read-default>
      <write-default>deny</write-default>
      <exec-default>deny</exec-default>
      <enable-external-groups>true</enable-external-groups>
      <rule-list>
        <name>admin</name>
        <group>PRIV15</group>
        <rule>
          <name>permit-all</name>
          <module-name>*</module-name>
          <access-operations>*</access-operations>
          <action>permit</action>
        </rule>
      </rule-list>
    </nacm>
  </data>
</rpc-reply>
```

The configuration that is pushed for NACM can withstand the reload of the router/switch, and it is part of the YANG DMI rather than the system’s running configuration, which would require the config to be saved to remain intact between reloads.

When augmented with user- and group-based rulesets, a role-based deployment construct is achievable. By default, the only user privilege level that is permitted to perform API operations is privilege level 15. Group mappings can be performed and mapped to align the respective user group, with the supported actions that are permitted for execution. Figure 23-7 shows an overview of how such mappings are applied.



Enable the read-only user with:

```
C9300(config)# username priv1 privilege 1 password priv1password
C9300# request platform software yang-management nacm populate-read-rules privilege 1
```

Privilege level	NACM group name
0	PRIV00
1	PRIV01
2	PRIV02
3	PRIV03
4	PRIV04
5	PRIV05
6	PRIV06
7	PRIV07
8	PRIV08
9	PRIV09
10	PRIV10
11	PRIV11
12	PRIV12
13	PRIV13
14	PRIV14
15	PRIV15

When specifying the read-only privilege level:

1. Allowed RPC's are set (get, get-config, get-scema)

2. Sensitive information is masked Native/enable, native/aaa, native/username

3. Read-only access to all models is provided

Default deny rule applied for all other operations and xpaths

Figure 23-7 *NAMC Mappings for User Privilege*

When privilege level 1 read-only rules are activated, users are able to retrieve the configuration from the system. However, secure items such as passwords are redacted for security reasons.

Note

At the time of this writing, although AAA-based authentication and authorization for the privilege level are possible in conjunction with NACM, it does not support dynamic provisioning of rulesets via TACACS+ or RADIUS. Instead, such provisioning must take place directly via NETCONF.

When you deploy Infrastructure as Code using network orchestrators such as Cisco Catalyst Center, Cisco ISE, or Cisco APIC, the CLI-based approaches that were demonstrated for provisioning are executed via the orchestrators themselves. The provisioning tasks that are performed through tools like Terraform are interfacing with the REST APIs of the orchestrators directly as a means to execute the provisioning functions. The selection of orchestrator-based versus device direct provisioning needs to be carefully considered by the network planning and operations teams to identify which method is the most scalable, manageable, and robust for their respective deployment.

Deploying a Resilient as Code Infrastructure

So far, we have shared various options around how to deploy configuration changes through network orchestrators using REST APIs, using NETCONF and RESTCONF securely, and safely to avoid losing connectivity with key network nodes. This approach works well for singular configurations, but networks are not singular node entities: They are multilayered and dependent infrastructure and software architectures that work in a cohesive manner to achieve optimal forwarding of IP traffic.

How does the deployment of changes in a network, along with its respective redundant components, take place while avoiding dependent execution steps that result in a mismatched future state? In many network environments, a legacy configuration of service-affecting changes would take place using console servers (if they exist) or simply by configuring a scheduled reload statement (**reload in 30**, for instance) to ensure that if something doesn’t go to plan, then recovery of the best-known previous state is possible.

In an “as code” network deployment, the deployment strives to move beyond the basic execution of a Python script, to go ahead and configure single nodes or potentially multiple nodes in batch deployment. In modern deployments, continuous integration/continuous deployment come into play, where an update is executed as a sequence of actions within a broader dependent task list. This approach can be applied through the use of a network orchestrator, which may include extra levels of sanity checking for configurations being applied or direct to device.

As is best practice within any software development environment, it is essential to maintain access to critical dependent components that are needed for the “as code” execution to take place. This means that the location of server instances for Git or Bitbucket, Jenkins or Gitlab, Robot/Pyats, or other testing frameworks must be deployed in the IT architecture in a manner that will support their reachability and align with the corporate service-level objectives.

“As Code” Today

“As code” provisioning has become ubiquitous with cloud native architectures, and for good reason: Hyperscalers like AWS, GCP, and Azure have taken an API-first approach to building out their cloud products and services. This approach is described as the development efforts around prototyping new services, functionality, and features happening first with the intent being mapped to API calls to achieve the needed system function, prior to a graphical user interface (UX) or a system CLI ever being created. The advantages of this approach are clear because the execution in code tends to be optimized to achieve the outcome in a limited number of API calls and steps, as opposed to a system that is GUI first, which often results in many layers of nested calls and identifiers that are required to achieve a specific task. Some of Cisco’s platforms do have well-optimized APIs to execute their intent. Unfortunately, this cannot be said for all of the platforms on the market today.

When interfacing with either a programmability-enabled system, such as ISE, or a network orchestrator, such as the cloud-based Meraki Dashboard, Cisco Catalyst SD-WAN Manager, or APIC, various options exist in terms of how that interaction takes place. While you can perform tasks on these systems using a Python script and execute calls via a REST API directly or via an SDK, without extended efforts, such an approach tends to miss tracking the state of the system and is usually quite intensive in terms of developer effort and overhead. Another challenge with such an approach is that building one-off Python scripts tends to result in future challenges in maintainability. It is not uncommon for a script that is maintained in an organization to have one primary author. Once that author leaves, the script becomes orphaned and lacks further maintenance within the organization.

To avoid such challenges, a number of Infrastructure as Code solutions, described earlier in this chapter, became available on the market, initially as open-source projects. Ansible, which was built up and heavily used in the system administrator domain, significantly lowered the entry level for individuals who want to either save time on programming or are not savvy with programming skills to perform provisioning tasks with relative ease.

In Ansible, a playbook is created, allowing the operator to create a simple, almost grocery-like list of tasks to be executed, resulting in simplified execution at scale and embedded monitorability.

Today, the legacy methods of provisioning are slowly becoming obsolete, with “as code” becoming the new default and best practice toward building scalable, reliable, and automatable architectures.

Transitioning to a Network “as Code”

Often customers ask the question, “What is the biggest challenge when shifting to Infrastructure as Code within an organization?” The answer to this question typically sits 30 centimeters from the keyboard. While there are technical challenges in terms of network standardization, the deployment of a viable and robust data model, the introduction of automation systems, the resilience of those systems, and the right observability being put into place, the largest challenge is, by far, the people. Figure 23-8 shows the journey that is often taken when moving from a legacy to an “as code” approach.

Over the years, many customer organizations have been subject to cost pressures from external markets, being compelled to do more with less. This approach unfortunately tends to result in the network planning and operations teams being in a perpetual fire-fighting mode of operations. They focus heavily on how to put out the next fire, remediate the next security threat, and work around best practices and designs based on a limited budget for the right design and the right hardware rather than spend the requisite time needed to transition to a new approach of doing things.

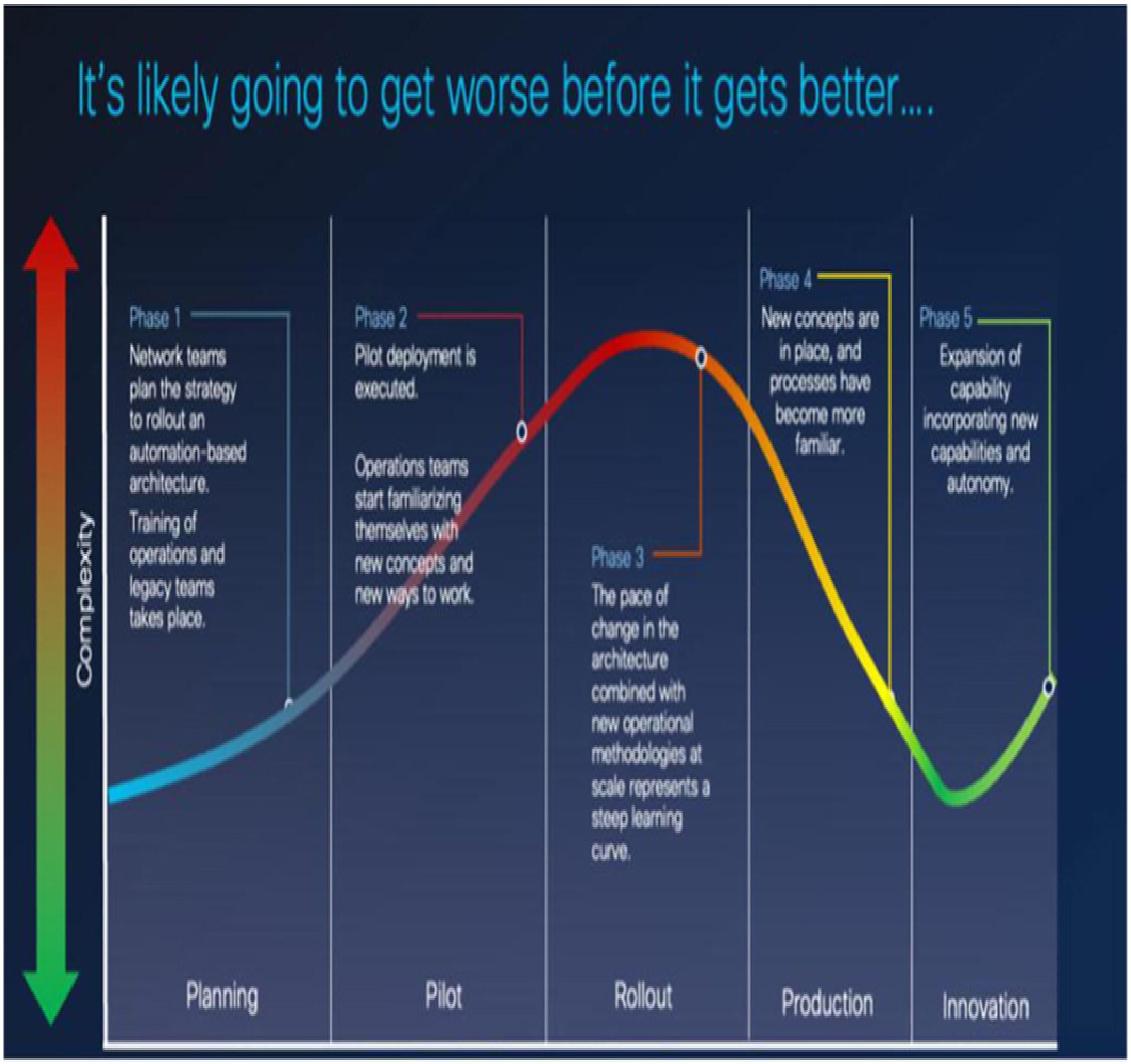


Figure 23-8 *Execution Path Observed in as Code Transitions*

Many network teams and specialists acknowledge that an "as code" approach is the North star in terms of where they should get to. However, a perpetual cycle of being overwhelmed in day-to-day activities tends to result in the goal posts constantly being shifted further and further away in terms of viable execution of this task. Because a change in the way that organizations need to operate is not simply a matter of flipping a switch or doing an update, buy-ins are generally needed from senior leadership, who need to shake things up to achieve a greater return in the mid-term rather than a tangible and measurable change overnight.

With time as network size grows, organizations that have shifted to an "as code" approach versus a legacy approach will observe that the trouble tickets, issues, and overhead do not grow in a linear fashion. Instead, the size of the network operations organization and the complexity it needs to deal with correlate with the quality of the automation and testing that is in place.

Pre-Validation in the Physical Replica or a Digital Twin

As we alluded to earlier, one of the key drivers around organizations shifting to automation is reducing overall operational expenditures. This goal is generally achievable through the transition to automation; however, it does not necessarily happen without a number of changes to the way that things are performed and standardized.

Reflecting on how the largest IT companies and service providers deploy their architectures today, standardization is key. This aligns to companies following frameworks such as ITIL and maintaining consistency above all else within their global infrastructure and software estates. An example is as rudimentary as port allocation on devices: The same port must always be used as the uplink from a CE to a PE device, and the same network module must always be used in a device for connectivity to ensure consistent numbering. Taking things further, having limited and consistent site types for deployments, with no more than five different variations, also ensures consistency when deploying globally and fostering future automation.

The advantage to following these principles is that fault identification and pre-validation exercises, including testing and automation, can be executed with an increased level of autonomy. Furthermore, more advanced fault triage activities, such predicting device failures before they happen, become more achievable when standardization is maintained.

Quite often, IT leadership teams, which may not be versed in the principles of automation, are reluctant to take such an approach, failing to invest in the right physical lab architectures that represent the site types that are deployed in production. Unfortunately, modern deployments that lack a viable physical and/or digital twin that can represent a comparable state to production environments come at a cost. High execution reliability and low downtime, which match with aspirations of cost reduction within an organization, can only be achieved in a viable pre-test and validation environment that is part of the precursor steps to production execution. Taking shortcuts here, from our experience with Cisco, tends only to lead to escalations and outages down the road.

Summary

Infrastructure as Code represents a key shift in the way that networks are managed, run, maintained, optimized, and improved. The network as the central autobahn that allows organizations to send critical business relevant information has only gained importance over the past decades, with size, scale, and complexity ever increasing.

As newer systems that focus more heavily on machine-to-machine communications become commonplace and as AI-based agents begin to engage in repetitive low-level troubleshooting and operational tasks that can be troubleshot frequently within a network estate, a shift to supervised administration of the network will become more common for certain tasks. This, under the circumstances, will shift the role of the network operator toward a better understanding of how copilots supporting network teams in troubleshooting tasks are coming to their conclusions and assessments for problems they encounter and delving deeper into the patterns that appear within their environments on a day-to-day basis. Many of these tasks are not possible without the right levels of automation, telemetry, and assessment that would happen if the network were not to be run “as code.”