Study
Planner

Review
Exercises

Official Cert Guide

Advance your IT career with hands-on learning

CCNP and CCIE Security Core

SCOR 350-701

OMAR SANTOS

ciscopress.com

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

CCNP and CCIE Security Core

SCOR 350-701

Official Cert Guide

OMAR SANTOS

Cisco Press
221 River St.
Hoboken, NJ 07030 USA

CCNP and CCIE Security Core SCOR 350-701 Official Cert Guide

Omar Santos

Published by:
Cisco Press
221 River St.
Hoboken, NJ 07030 USA

ScoutAutomatedPrintCode

Library of Congress Control Number: 2020901233

ISBN-10: 0-13-597197-7

ISBN-13: 978-0-13-597197-0

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.

Editor-in-Chief: Mark Taub

Alliances Manager, Cisco Press: Arezou Gol

Director, Product Manager: Brett Bartow

Managing Editor: Sandra Schroeder

Development Editor: Christopher A. Cleveland

Project Editor: Mandie Frank

Copy Editor: Bart Reed

Technical Editor: John Stuppi

Editorial Assistant: Cindy Teeters

Designer: Chuti Prasertsith

Composition: codeMantra

Indexer: Ken Johnson

Proofreader: Abigail Manheim

Credits

Figure 1-1	Screenshot of The Exploit Database (Exploit-DB) © OffSec Services Limited 2020
Figure 1-2	Screenshot of Using searchsploit © OffSec Services Limited 2020
Figure 1-4	Screenshot of Ghidra Software Reverse Engineering Framework, ghidra
Figure 1-6	Screenshot of SQL injection vulnerability © Webgoat SQL Injection
Figure 3-27	Screenshot of Installing the Python requests package using pip © Python Software Foundation
Figure 3-28	Screenshot of Using the Python requests package © Python Software Foundation
Figure 3-29	Screenshot of Using curl to obtain information from an API © GitHub, Inc.
Figure 3-30	Screenshot of Using curl to obtain additional information from the Deck of Cards API © GitHub, Inc.
Figure 9-11	Screenshot of AWS Lamda © 2020, Amazon Web Services, Inc
Figure 9-14	Screenshot of Docker © 2020 Docker Inc.
Figure 9-15	Screenshot of Docker © 2020 Docker Inc.
Figure 9-16	Screenshot of Docker © 2020 Docker Inc.
Figure 9-17	Deploying your first app on Kubernetes, Google Inc.
Figure 9-19	Screenshot of The Kubernetes Authors © Google Inc.
Figure 9-20	Screenshot of The Kubernetes Authors © Google Inc.
Figure 9-21	Screenshot of The Kubernetes Authors © Google Inc.
Figure 10-2	Screenshot of macOS © Apple 2019

The International Organization for Standardization (ISO), ISO/IEC 27001:2005(en)

The International Organization for Standardization (ISO)

Malware Tunneling in IPv6, June 22, 2012. United States Department of Homeland Security

The International Organization for Standardization (ISO)

NIST Special Publication 800-61

NIST Special Publication 800-61

NIST Special Publication 800-61

NIST Special Publication 800-61

US-CERT Description Document - RFC 2350

Cybersecurity and Infrastructure Security Agency (CISA), U.S. Department of Homeland Security

NIST Special Publication 800-63B

Contents at a Glance

Contents

About the Author

Omar Santos is an active member of the security community, where he leads several industry-wide initiatives and standard bodies. His active role helps businesses, academic institutions, state and local law enforcement agencies, and other participants dedicated to increasing the security of the critical infrastructure.

Omar is the author of more than 20 books and video courses as well as numerous white papers, articles, and security configuration guidelines and best practices. Omar is a Principal Engineer of the Cisco Product Security Incident Response Team (PSIRT), where he mentors and leads engineers and incident managers during the investigation and resolution of security vulnerabilities.

Omar has been quoted by numerous media outlets, such as TheRegister, Wired, ZDNet, ThreatPost, CyberScoop, TechCrunch, Fortune Magazine, Ars Technica, and more. You can follow Omar on Twitter @santosomar.

About the Technical Reviewer

John Stuppi, CCIE No. 11154, is a Technical Leader in the Customer Experience Security Programs (CXSP) organization at Cisco where he consults with Cisco customers on protecting their networks against existing and emerging cyber security threats, risks, and vulnerabilities. Current projects include working with newly acquired entities to integrate them into the Cisco PSIRT Vulnerability Management processes. John has presented multiple times on various network security topics at Cisco Live, Black Hat, as well as other customer-facing cyber security conferences. John is also the co-author of the *Official Certification Guide for CCNA Security 210-260* published by Cisco Press. Additionally, John has contributed to the Cisco Security Portal through the publication of white papers, Security Blog posts, and Cyber Risk Report articles. Prior to joining Cisco, John worked as a network engineer for JPMorgan, and then as a network security engineer at Time, Inc., with both positions based in New York City. John is also a CISSP (No. 25525) and holds AWS Cloud Practitioner and Information Systems Security (INFOSEC) Professional Certifications. In addition, John has a BSEE from Lehigh University and an MBA from Rutgers University. John lives in Ocean Township, New Jersey (down on the “Jersey Shore”) with his wife, two kids, and his dog.

Dedication

I would like to dedicate this book to my lovely wife, Jeannette, and my two beautiful children, Hannah and Derek, who have inspired and supported me throughout the development of this book.

Acknowledgments

I would like to thank the technical editor and my good friend, John Stuppi, for his time and technical expertise.

I would like to thank the Cisco Press team, especially James Manly and Christopher Cleveland, for their patience, guidance, and consideration.

Finally, I would like to thank Cisco and the Cisco Product Security Incident Response Team (PSIRT), Security Research, and Operations for enabling me to constantly learn and achieve many goals throughout all these years.

Introduction

The Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) exam is the required "core" exam for the CCNP Security and CCIE Security certifications. If you pass the SCOR 350-701 exam, you also obtain the Cisco Certified Specialist – Security Core Certification. This exam covers core security technologies, including cybersecurity fundamentals, network security, cloud security, identity management, secure network access, endpoint protection and detection, and visibility and enforcement.

The Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) is a 120-minute exam.

TIP You can review the exam blueprint from Cisco's website at https://learningnetwork.cisco.com/community/certifications/ccnp-security/scor/exam-topics.

This book gives you the foundation and covers the topics necessary to start your CCNP Security or CCIE Security journey.

The CCNP Security Certification

The CCNP Security certification is one of the industry's most respected certifications. In order for you to earn the CCNP Security certification, you must pass two exams: the SCOR exam covered in this book (which covers core security technologies) and one security concentration exam of your choice, so you can customize your certification to your technical area of focus.

TIP The SCOR core exam is also the qualifying exam for the CCIE Security certification. Passing this exam is the first step toward earning both of these certifications.

The following are the CCNP Security concentration exams:

- Securing Networks with Cisco Firepower (SNCF 300-710)
- Implementing and Configuring Cisco Identity Services Engine (SISE 300-715)
- Securing Email with Cisco Email Security Appliance (SESA 300-720)
- Securing the Web with Cisco Web Security Appliance (SWSA 300-725)
- Implementing Secure Solutions with Virtual Private Networks (SVPN 300-730)
- Automating Cisco Security Solutions (SAUTO 300-735)

TIP CCNP Security now includes automation and programmability to help you scale your security infrastructure. If you pass the Developing Applications Using Cisco Core Platforms and APIs v1.0 (DEVCOR 350-901) exam, the SCOR exam, and the Automating Cisco Security Solutions (SAUTO 300-735) exam, you will achieve the CCNP Security and DevNet Professional certifications with only three exams. Every exam earns an individual Specialist certification, allowing you to get recognized for each of your accomplishments, instead of waiting until you pass all the exams.

There are no formal prerequisites for CCNP Security. In other words, you do not have to pass the CCNA Security or any other certifications in order to take CCNP-level exams. The same goes for the CCIE exams. On the other hand, CCNP candidates often have three to five years of experience in IT and cybersecurity.

Cisco considers ideal candidates to be those that possess the following:

- Knowledge of implementing and operating core security technologies
- Understanding of cloud security
- Hands-on experience with next-generation firewalls, intrusion prevention systems (IPSs), and other network infrastructure devices
- Understanding of content security, endpoint protection and detection, and secure network access, visibility, and enforcement
- Understanding of cybersecurity concepts with hands-on experience in implementing security controls

The CCIE Security Certification

The CCIE Security certification is one of the most admired and elite certifications in the industry. The CCIE Security program prepares you to be a recognized technical leader. In order to earn the CCIE Security certification, you must pass the SCOR 350-701 exam and an 8-hour, hands-on lab exam. The lab exam covers very complex network security scenarios. These scenarios range from designing through deploying, operating, and optimizing security solutions.

Cisco considers ideal candidates to be those who possess the following:

- Extensive hands-on experience with Cisco's security portfolio
- Experience deploying Cisco's next-generation firewalls and next-generation IPS devices
- Deep understanding of secure connectivity and segmentation solutions
- Hands-on experience with infrastructure device hardening and infrastructure security
- Configuring and troubleshooting identity management, information exchange, and access control
- Deep understanding of advanced threat protection and content security

The Exam Objectives (Domains)

The Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) exam is broken down into six major domains. The contents of this book cover each of the domains and the subtopics included in them, as illustrated in the following descriptions.

The following table breaks down each of the domains represented in the exam.

Domain	Percentage of Representation in Exam
1: Security Concepts	25%
2: Network Security	20%
3: Securing the Cloud	15%
4: Content Security	15%
5: Endpoint Protection and Detection	10%
6: Secure Network Access, Visibility, and Enforcement	15%
	Total 100%

Here are the details of each domain:

Domain 1: Monitoring and Reporting: This domain is covered in Chapters 1, 2, 3, and 8.

1.1 Explain common threats against on-premises and cloud environments

1.1.a On-premises: viruses, trojans, DoS/DDoS attacks, phishing, rootkits, man-in-the-middle attacks, SQL injection, cross-site scripting, malware

1.1.b Cloud: data breaches, insecure APIs, DoS/DDoS, compromised credentials

1.2 Compare common security vulnerabilities such as software bugs, weak and/or hardcoded passwords, SQL injection, missing encryption, buffer overflow, path traversal, cross-site scripting/forgery

1.3 Describe functions of the cryptography components such as hashing, encryption, PKI, SSL, IPsec, NAT-T IPv4 for IPsec, pre-shared key, and certificate-based authorization

1.4 Compare site-to-site VPN and remote access VPN deployment types such as sVTI, IPsec, Cryptomap, DMVPN, FLEXVPN, including high availability considerations, and AnyConnect

1.5 Describe security intelligence authoring, sharing, and consumption

1.6 Explain the role of the endpoint in protecting humans from phishing and social engineering attacks

1.7 Explain northbound and southbound APIs in the SDN architecture

1.8 Explain DNAC APIs for network provisioning, optimization, monitoring, and troubleshooting

1.9 Interpret basic Python scripts used to call Cisco Security appliances APIs

Domain 2: Network Security: This domain is covered primarily in Chapters 5, 6, and 7.

2.1 Compare network security solutions that provide intrusion prevention and firewall capabilities

2.2 Describe deployment models of network security solutions and architectures that provide intrusion prevention and firewall capabilities

2.3 Describe the components, capabilities, and benefits of NetFlow and Flexible NetFlow records

2.4 Configure and verify network infrastructure security methods (router, switch, wireless)

2.4.a Layer 2 methods (network segmentation using VLANs and VRF-lite; Layer 2 and port security; DHCP snooping; dynamic ARP inspection; storm control; PVLANs to segregate network traffic; and defenses against MAC, ARP, VLAN hopping, STP, and DHCP rogue attacks)

2.4.b Device hardening of network infrastructure security devices (control plane, data plane, management plane, and routing protocol security)

2.5 Implement segmentation, access control policies, AVC, URL filtering, and malware protection

2.6 Implement management options for network security solutions such as intrusion prevention and perimeter security (single vs. multidevice manager, in-band vs. out-of-band, CDP, DNS, SCP, SFTP, and DHCP security and risks)

2.7 Configure AAA for device and network access (authentication and authorization, TACACS+, RADIUS and RADIUS flows, accounting, and dACL)

2.8 Configure secure network management of perimeter security and infrastructure devices (secure device management, SNMPv3, views, groups, users, authentication, encryption, secure logging, and NTP with authentication)

2.9 Configure and verify site-to-site VPN and remote access VPN

2.9.a Site-to-site VPN utilizing Cisco routers and IOS

2.9.b Remote access VPN using Cisco AnyConnect Secure Mobility client

2.9.c Debug commands to view IPsec tunnel establishment and troubleshooting

Domain 3: Securing the Cloud: This domain is covered primarily in Chapter 9.

3.1 Identify security solutions for cloud environments

3.1.a Public, private, hybrid, and community clouds

3.1.b Cloud service models: SaaS, PaaS, and IaaS (NIST 800-145)

3.2 Compare the customer vs. provider security responsibility for the different cloud service models

3.2.a Patch management in the cloud

3.2.b Security assessment in the cloud

   3.2.c Cloud-delivered security solutions such as firewall, management, proxy, security intelligence, and CASB

- 3.3 Describe the concept of DevSecOps (CI/CD pipeline, container orchestration, and security)
- 3.4 Implement application and data security in cloud environments
- 3.5 Identify security capabilities, deployment models, and policy management to secure the cloud
- 3.6 Configure cloud logging and monitoring methodologies
- 3.7 Describe application and workload security concepts

Domain 4: Content Security: This domain is covered primarily in Chapter 10.

- 4.1 Implement traffic redirection and capture methods
- 4.2 Describe web proxy identity and authentication, including transparent user identification
- 4.3 Compare the components, capabilities, and benefits of local and cloud-based email and web solutions (ESA, CES, WSA)
- 4.4 Configure and verify web and email security deployment methods to protect on-premises and remote users (inbound and outbound controls and policy management)
- 4.5 Configure and verify email security features such as SPAM filtering, antimalware filtering, DLP, blacklisting, and email encryption
- 4.6 Configure and verify secure Internet gateway and web security features such as blacklisting, URL filtering, malware scanning, URL categorization, web application filtering, and TLS decryption
- 4.7 Describe the components, capabilities, and benefits of Cisco Umbrella
- 4.8 Configure and verify web security controls on Cisco Umbrella (identities, URL content settings, destination lists, and reporting)

Domain 5: Endpoint Protection and Detection: This domain is covered primarily in Chapter 11.

- 5.1 Compare Endpoint Protection Platforms (EPPs) and Endpoint Detection & Response (EDR) solutions
- 5.2 Explain antimalware, retrospective security, Indication of Compromise (IOC), antivirus, dynamic file analysis, and endpoint-sourced telemetry
- 5.3 Configure and verify outbreak control and quarantines to limit infection
- 5.4 Describe justifications for endpoint-based security
- 5.5 Describe the value of endpoint device management and asset inventory such as MDM

5.6 Describe the uses and importance of a multifactor authentication (MFA) strategy

5.7 Describe endpoint posture assessment solutions to ensure endpoint security

5.8 Explain the importance of an endpoint patching strategy

Domain 6: Secure Network Access, Visibility, and Enforcement: This domain is covered primarily in Chapters 4 and 5.

6.1 Describe identity management and secure network access concepts such as guest services, profiling, posture assessment, and BYOD

6.2 Configure and verify network access device functionality such as 802.1X, MAB, and WebAuth

6.3 Describe network access with CoA

6.4 Describe the benefits of device compliance and application control

6.5 Explain exfiltration techniques (DNS tunneling, HTTPS, email, FTP/SSH/SCP/SFTP, ICMP, Messenger, IRC, and NTP)

6.6 Describe the benefits of network telemetry

6.7 Describe the components, capabilities, and benefits of these security products and solutions:

6.7.a Cisco Stealthwatch

6.7.b Cisco Stealthwatch Cloud

6.7.c Cisco pxGrid

6.7.d Cisco Umbrella Investigate

6.7.e Cisco Cognitive Threat Analytics

6.7.f Cisco Encrypted Traffic Analytics

6.7.g Cisco AnyConnect Network Visibility Module (NVM)

Steps to Pass the SCOR Exam

There are no prerequisites for the SCOR exam. However, students must have an understanding of networking and cybersecurity concepts.

Signing Up for the Exam

The steps required to sign up for the SCOR exam as follows:

1. Create an account at https://home.pearsonvue.com/cisco.
2. Complete the Examination Agreement, attesting to the truth of your assertions regarding professional experience and legally committing to the adherence of the testing policies.
3. Submit the examination fee.

Facts About the Exam

The exam is a computer-based test. The exam consists of multiple-choice questions only. You must bring a government-issued identification card. No other forms of ID will be accepted.

TIP Refer to the Cisco Certification site at https://cisco.com/go/certifications for more information regarding this, and other, Cisco certifications.

About the CCNP and CCIE Security Core SCOR 350-701 Official Cert Guide

This book maps directly to the topic areas of the SCOR exam and uses a number of features to help you understand the topics and prepare for the exam.

Objectives and Methods

This book uses several key methodologies to help you discover the exam topics that need more review, to help you fully understand and remember those details, and to help you prove to yourself that you have retained your knowledge of those topics. This book does not try to help you pass the exam only by memorization; it seeks to help you to truly learn and understand the topics. This book is designed to help you pass the Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) exam by using the following methods:

- Helping you discover which exam topics you have not mastered
- Providing explanations and information to fill in your knowledge gaps
- Supplying exercises that enhance your ability to recall and deduce the answers to test questions
- Providing practice exercises on the topics and the testing process via test questions on the companion website

Book Features

To help you customize your study time using this book, the core chapters have several features that help you make the best use of your time:

- **Foundation Topics:** These are the core sections of each chapter. They explain the concepts for the topics in that chapter.
- **Exam Preparation Tasks:** After the "Foundation Topics" section of each chapter, the "Exam Preparation Tasks" section lists a series of study activities that you should do at the end of the chapter:
  - **Review All Key Topics:** The Key Topic icon appears next to the most important items in the "Foundation Topics" section of the chapter. The Review All Key Topics activity lists the key topics from the chapter, along with their page numbers. Although the contents of the entire chapter could be on the exam, you should definitely know the information listed in each key topic, so you should review these.

  - **Define Key Terms:** Although the Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) exam may be unlikely to ask a question such as "Define this term," the exam does require that you learn and know a lot of cybersecurity terminology. This section lists the most important terms from the chapter, asking you to write a short definition and compare your answer to the glossary at the end of the book.
  - **Review Questions:** Confirm that you understand the content you just covered by answering these questions and reading the answer explanations.
- **Web-based practice exam:** The companion website includes the Pearson Cert Practice Test engine, which allows you to take practice exam questions. Use it to prepare with a sample exam and to pinpoint topics where you need more study.

How This Book Is Organized

This book contains 11 core chapters—Chapters 1 through 11. Chapter 12 includes preparation tips and suggestions for how to approach the exam. Each core chapter covers a subset of the topics on the Implementing and Operating Cisco Security Core Technologies (SCOR 350-701) exam. The core chapters map to the SCOR topic areas and cover the concepts and technologies you will encounter on the exam.

The Companion Website for Online Content Review

All the electronic review elements, as well as other electronic components of the book, exist on this book's companion website.

To access the companion website, which gives you access to the electronic content with this book, start by establishing a login at www.ciscopress.com and registering your book.

To do so, simply go to www.ciscopress.com/register and enter the ISBN of the print book: 9780135971970. After you have registered your book, go to your account page and click the **Registered Products** tab. From there, click the **Access Bonus Content** link to get access to the book's companion website.

Note that if you buy the *Premium Edition eBook and Practice Test* version of this book from Cisco Press, your book will automatically be registered on your account page. Simply go to your account page, click the **Registered Products** tab, and select **Access Bonus Content** to access the book's companion website.

Please note that many of our companion content files can be very large, especially image and video files.

If you are unable to locate the files for this title by following the steps at left, please visit www.pearsonITcertification.com/contact and select the **Site Problems/Comments** option. Our customer service representatives will assist you.

NOTE Other eBook customers: As of the time of publication, only the publisher and Amazon supply PTP access codes when you purchase their eBook editions of this book.

Customizing Your Exams

Once you are in the exam settings screen, you can choose to take exams in one of three modes:

- **Study mode:** Allows you to fully customize your exams and review answers as you are taking the exam. This is typically the mode you would use first to assess your knowledge and identify information gaps.
- **Practice Exam mode:** Locks certain customization options, as it is presenting a realistic exam experience. Use this mode when you are preparing to test your exam readiness.
- **Flash Card mode:** Strips out the answers and presents you with only the question stem. This mode is great for late-stage preparation when you really want to challenge yourself to provide answers without the benefit of seeing multiple-choice options. This mode does not provide the detailed score reports that the other two modes do, so you should not use it if you are trying to identify knowledge gaps.

In addition to these three modes, you will be able to select the source of your questions. You can choose to take exams that cover all of the chapters or you can narrow your selection to just a single chapter or the chapters that make up specific parts in the book. All chapters are selected by default. If you want to narrow your focus to individual chapters, simply deselect all the chapters and then select only those on which you wish to focus in the Objectives area.

You can also select the exam banks on which to focus. Each exam bank comes complete with a full exam of questions that cover topics in every chapter. The two exams printed in the book are available to you as well as two additional exams of unique questions. You can have the test engine serve up exams from all four banks or just from one individual bank by selecting the desired banks in the exam bank area.

There are several other customizations you can make to your exam from the exam settings screen, such as the time of the exam, the number of questions served up, whether to randomize questions and answers, whether to show the number of correct answers for multiple-answer questions, and whether to serve up only specific types of questions. You can also create custom test banks by selecting only questions that you have marked or questions on which you have added notes.

Updating Your Exams

If you are using the online version of the Pearson Test Prep software, you should always have access to the latest version of the software as well as the exam data. If you are using the Windows desktop version, every time you launch the software while connected to the Internet, it checks if there are any updates to your exam data and automatically downloads any changes that were made since the last time you used the software.

Sometimes, due to many factors, the exam data may not fully download when you activate your exam. If you find that figures or exhibits are missing, you may need to manually update your exams. To update a particular exam you have already activated and downloaded, simply click the **Tools** tab and click the **Update Products** button. Again, this is only an issue with the desktop Windows application.

If you wish to check for updates to the Pearson Test Prep exam engine software, Windows desktop version, simply click the **Tools** tab and click the **Update Application** button. This ensures that you are running the latest version of the software engine.

CHAPTER 3

Software-Defined Networking Security and Network Programmability

This chapter covers the following topics:

Software-Defined Networking (SDN) and SDN Security

Network Programmability

This chapter starts with an introduction to SDN and different SDN security concepts, such as centralized policy management and micro-segmentation. This chapter also introduces SDN solutions such as Cisco ACI and modern networking environments such as Cisco DNA. You will also learn what are network overlays and what they are trying to solve.

The second part of this chapter provides an overview of network programmability and how networks are being managed using modern application programming interfaces (APIs) and other functions. This chapter also includes dozens of references that are available to enhance your learning.

The following SCOR 350-701 exam objectives are covered in this chapter:

- **Domain 1: Security Concepts**
  - 1.7 Explain northbound and southbound APIs in the SDN architecture
  - 1.8 Explain DNAC APIs for network provisioning, optimization, monitoring, and troubleshooting

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the "Exam Preparation Tasks" section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 3-1 lists the major headings in this chapter and their corresponding "Do I Know This Already?" quiz questions. You can find the answers in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Q&A Sections."

Table 3-1 "Do I Know This Already?" Section-to-Question Mapping

Foundation Topics Section	Questions
Software-Defined Networking (SDN) and SDN Security	1–5
Network Programmability	6–10

CAUTION The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you incorrectly guess skews your self-assessment results and might provide you with a false sense of security.

1. Which of the following are the three different "planes" in traditional networking?
   a. The management, control, and data planes
   b. The authorization, authentication, and accountability planes
   c. The authentication, control, and data planes
   d. None of these answers is correct.
2. Which of the following is true about Cisco ACI?
   a. Spine nodes interconnect leaf devices, and they can also be used to establish connections from a Cisco ACI pod to an IP network or interconnect multiple Cisco ACI pods.
   b. Leaf switches provide the Virtual Extensible LAN (VXLAN) tunnel endpoint (VTEP) function.
   c. The APIC manages the distributed policy repository responsible for the definition and deployment of the policy-based configuration of the Cisco ACI infrastructure.
   d. All of these answers are correct.
3. Which of the following is used to create network overlays?
   a. SDN-Lane
   b. VXLAN
   c. VXWAN
   d. None of these answers is correct.
4. Which of the following is an identifier or a tag that represents a logical segment?
   a. VXLAN Network Identifier (VNID)
   b. VXLAN Segment Identifier (VSID)
   c. ACI Network Identifier (ANID)
   d. Application Policy Infrastructure Controller (APIC)
5. Which of the following is network traffic between servers (virtual servers or physical servers), containers, and so on?
   a. East-west traffic
   b. North-south traffic
   c. Micro-segmentation
   d. Network overlays

6. Which of the following is an HTTP status code message range related to successful HTTP transactions?
   a. Messages in the 100 range
   b. Messages in the 200 range
   c. Messages in the 400 range
   d. Messages in the 500 range
7. Which of the following is a Python package that can be used to interact with REST APIs?
   a. argparse
   b. requests
   c. rest_api_pkg
   d. None of these answers is correct.
8. Which of the following is a type of API that exclusively uses XML?
   a. APIC
   b. REST
   c. SOAP
   d. GraphQL
9. Which of the following is a modern framework of API documentation and is now the basis of the OpenAPI Specification (OAS)?
   a. SOAP
   b. REST
   c. Swagger
   d. WSDL
10. Which of the following can be used to retrieve a network device configuration?
   a. RESTCONF
   b. NETCONF
   c. SNMP
   d. All of these answers are correct.

Foundation Topics

Introduction to Software-Defined Networking

In the last decade there have been several shifts in networking technologies. Some of these changes are due to the demand of modern applications in very diverse environments and the cloud. This complexity introduces risks, including network configuration errors that can cause significant downtime and network security challenges.

Subsequently, networking functions such as routing, optimization, and security have also changed. The next generation of hardware and software components in enterprise networks must support both the rapid introduction and the rapid evolution of new technologies and solutions. Network infrastructure solutions must keep pace with the business environment and support modern capabilities that help drive simplification within the network.

These elements have fueled the creation of software-defined networking (SDN). SDN was originally created to decouple control from the forwarding functions in networking equipment. This is done to use software to centrally manage and "program" the hardware and virtual networking appliances to perform forwarding.

Key Topic

Traditional Networking Planes

In traditional networking, there are three different "planes" or elements that allow network devices to operate: the management, control, and data planes. Figure 3-1 shows a high-level explanation of each of the planes in traditional networking.

TRADITIONAL ROUTING AND SWITCHING PLANES		
Management Plane	Control Plane	Data Plane
• Configuration and monitoring • Typically done via the traditional CLI or GUI • Each vendor has its proprietary way to configure its devices	• Layer 2 protocols and control • Layer 3 protocols (e.g., OSPF, RIP, BGP, etc.)	• Institutes how data is forwarded inside the hardware from interface to interface

Figure 3-1 *The Management, Control, and Data Planes*

The control plane has always been separated from the data plane. There was no central brain (or controller) that controlled the configuration and forwarding. Let's take a look at the example shown in Figure 3-2. Routers, switches, and firewalls were managed by the command-line interface (CLI), graphical user interfaces (GUIs), and custom Tcl scripts. For instance, the firewalls were managed by the Adaptive Security Device Manager (ASDM), while the routers were managed by the CLI.

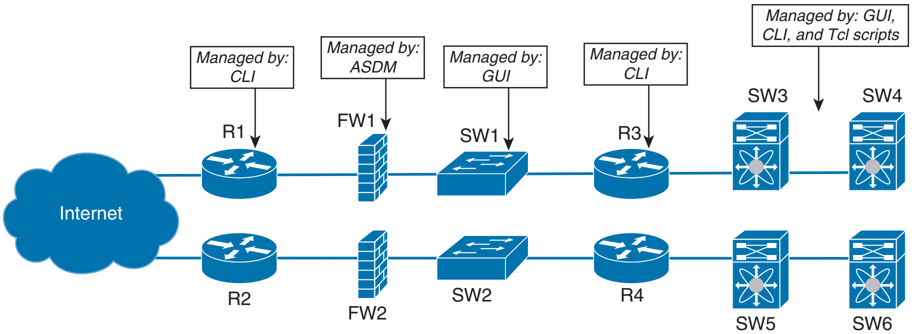


Figure 3-2 *Traditional Network Management Solutions*

Each device in Figure 3-2 has its "own brain" and does not really exchange any intelligent information with the rest of the devices.

Key Topic

So What's Different with SDN?

SDN introduced the notion of a centralized controller. The SDN controller has a global view of the network, and it uses a common management protocol to configure the network infrastructure devices. The SDN controller can also calculate reachability information from many systems in the network and pushes a set of flows inside the switches. The flows are used by the hardware to do the forwarding. Here you can see a clear transition from a distributed "semi-intelligent brain" approach to a "central and intelligent brain" approach.

TIP An example of an open source implementation of SDN controllers is the Open vSwitch (OVS) project using the OVS Database (OVSDB) management protocol and the OpenFlow protocol. Another example is the Cisco Application Policy Infrastructure Controller (Cisco APIC). Cisco APIC is the main architectural component and the brain of the Cisco Application Centric Infrastructure (ACI) solution. A great example of this is Cisco ACI, which is discussed in the next section of the chapter.

SDN changed a few things in the management, control, and data planes. However, the big change was in the control and data planes in software-based switches and routers (including virtual switches inside of hypervisors). For instance, the Open vSwitch project started some of these changes across the industry.

SDN provides numerous benefits in the area of management plane. These benefits are in both physical switches and virtual switches. SDN is now widely adopted in data centers. A great example of this is Cisco ACI.

Key Topic

Introduction to the Cisco ACI Solution

Cisco ACI provides the ability to automate setting networking policies and configurations in a very flexible and scalable way. Figure 3-3 illustrates the concept of a centralized policy and configuration management in the Cisco ACI solution.

The Cisco ACI scenario shown in Figure 3-3 uses a leaf-and-spine topology. Each leaf switch is connected to every spine switch in the network with no interconnection between leaf switches or spine switches.

The leaf switches have ports connected to traditional Ethernet devices (for example, servers, firewalls, routers, and so on). Leaf switches are typically deployed at the edge of the fabric. These leaf switches provide the Virtual Extensible LAN (VXLAN) tunnel endpoint (VTEP) function. VXLAN is a network virtualization technology that leverages an encapsulation technique (similar to VLANs) to encapsulate Layer 2 Ethernet frames within UDP packets (over UDP port 4789, by default).

NOTE The section "VXLAN and Network Overlays," later in the chapter, will discuss VXLAN and overlays in more detail.

In Cisco ACI, the IP address that represents the leaf VTEP is called the physical tunnel endpoint (PTEP). The leaf switches are responsible for routing or bridging tenant packets and for applying network policies.

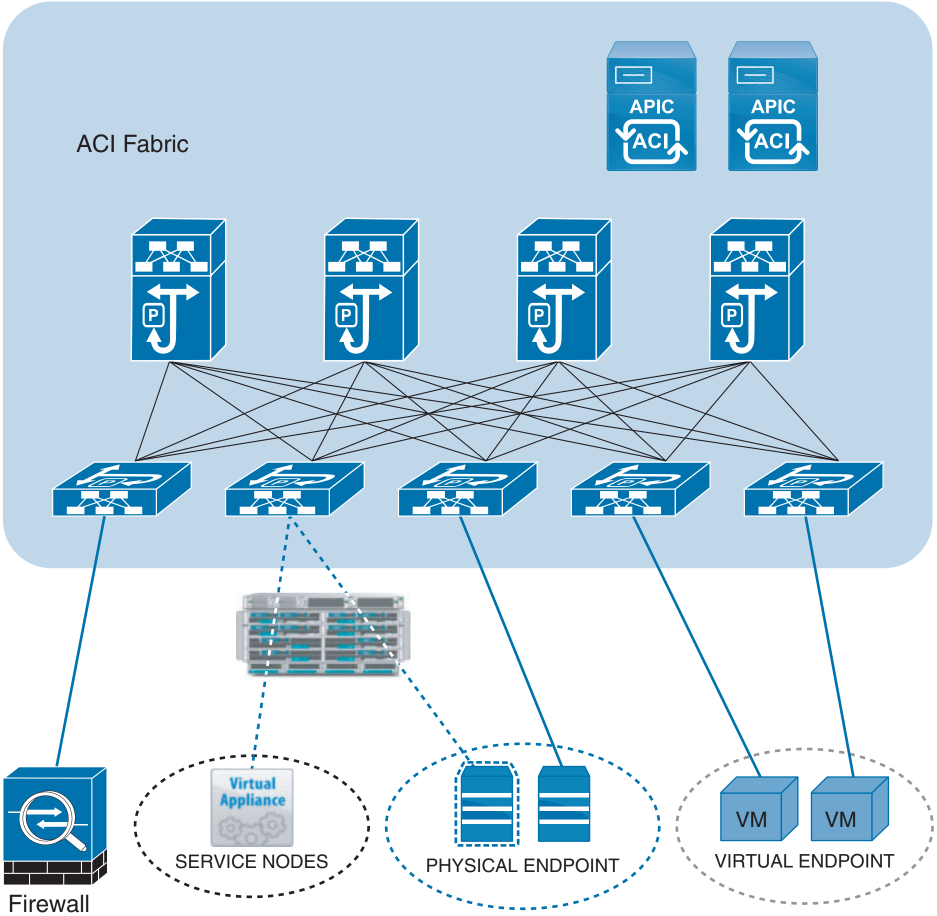


Figure 3-3 *Cisco APIC Configuration and Policy Management*

Spine nodes interconnect leaf devices, and they can also be used to establish connections from a Cisco ACI pod to an IP network or to interconnect multiple Cisco ACI pods. Spine switches store all the endpoint-to-VTEP mapping entries. All leaf nodes connect to all spine nodes within a Cisco ACI pod. However, no direct connectivity is allowed between spine nodes or between leaf nodes.

NOTE All workloads in Cisco ACI connect to leaf switches. The leaf switches used in a Cisco ACI fabric are Top-of-the-Rack (ToR) switches. The acronym "ToR" here is not the same as "The Onion Router" (a solution used for anonymity and to access the "deep web").

The APIC can be considered a policy and a topology manager. APIC manages the distributed policy repository responsible for the definition and deployment of the policy-based configuration of the Cisco ACI infrastructure. APIC also manages the topology and inventory information of all devices within the Cisco ACI pod.

Key Topic

The following are additional functions of the APIC:

- The APIC "observer" function monitors the health, state, and performance information of the Cisco ACI pod.
- The "boot director" function is in charge of the booting process and firmware updates of the spine switches, leaf switches, and the APIC components.
- The "appliance director" APIC function manages the formation and control of the APIC appliance cluster.
- The "virtual machine manager (VMM)" is an agent between the policy repository and a hypervisor. The VMM interacts with hypervisor management systems (for example, VMware vCenter).
- The "event manager" manages and stores all the events and faults initiated from the APIC and the Cisco ACI fabric nodes.
- The "appliance element" maintains the inventory and state of the local APIC appliance.

TIP The Cisco ACI Design Guide provides comprehensive information about the design, deployment, and configuration of the ACI solution. The design guide can be found here: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-737909.pdf.

Key Topic

VXLAN and Network Overlays

Modern networks and data centers need to provide load balancing, better scalability, elasticity, and faster convergence. Many organizations use the overlay network model. Deploying an overlay network allows you to tunnel Layer 2 Ethernet packets with different encapsulations over a Layer 3 network. The overlay network uses "tunnels" to carry the traffic across the Layer 3 fabric. This solution also needs to allow the "underlay" to separate network flows between different "tenants" (administrative domains). The solution also needs to switch packets within the same Layer 2 broadcast domain, route traffic between Layer 3 broadcast domains, and provide IP separation, traditionally done via virtual routing and forwarding (VRF).

There have been multiple IP tunneling mechanisms introduced throughout the years. The following are a few examples of tunneling mechanisms:

- Virtual Extensible LAN (VXLAN)
- Network Virtualization using Generic Routing Encapsulation (NVGRE)
- Stateless Transport Tunneling (STT)
- Generic Network Virtualization Encapsulation (GENEVE)

All of the aforementioned tunneling protocols carry an Ethernet frame inside an IP frame. The main difference between them is in the type of the IP frame used. For instance, VXLAN uses UDP, and STT uses TCP.

The use of UDP in VXLAN enables routers to apply hashing algorithms on the outer UDP header to load balance network traffic. Network traffic that is riding the overlay network tunnels is load balanced over multiple links using equal-cost multi-path routing (ECMP). This introduces a better solution compared to traditional network designs. In traditional network designs, access switches connect to distribution switches. This causes redundant links to block due to spanning tree.

VXLAN uses an identifier or a tag that represents a logical segment that is called the VXLAN Network Identifier (VNID). The logical segment identified with the VNID is a Layer 2 broadcast domain that is tunneled over the VTEP tunnels.

Figure 3-4 shows an example of an overlay network that provides Layer 2 capabilities.

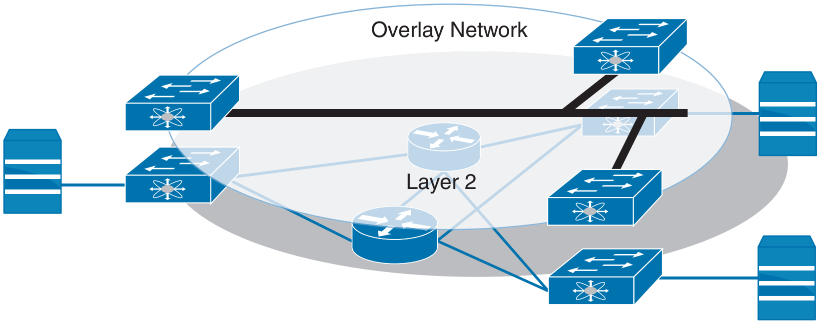


Figure 3-4 *Overlay Network Providing Layer 2 Capabilities*

Figure 3-5 shows an example of an overlay network that provides Layer 3 routing capabilities.

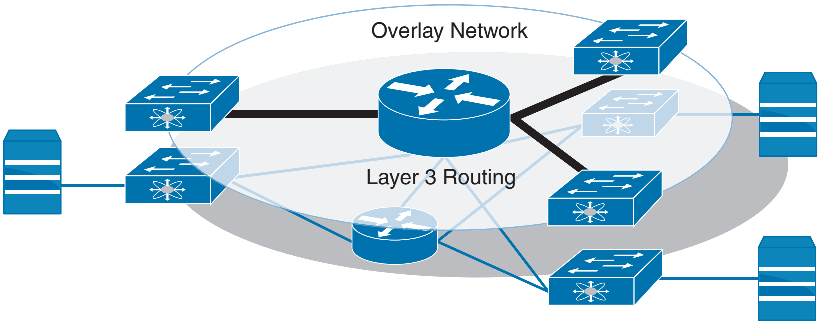


Figure 3-5 *Overlay Network Providing Layer 3 Routing Capabilities*

Figure 3-6 illustrates the VXLAN frame format for your reference.

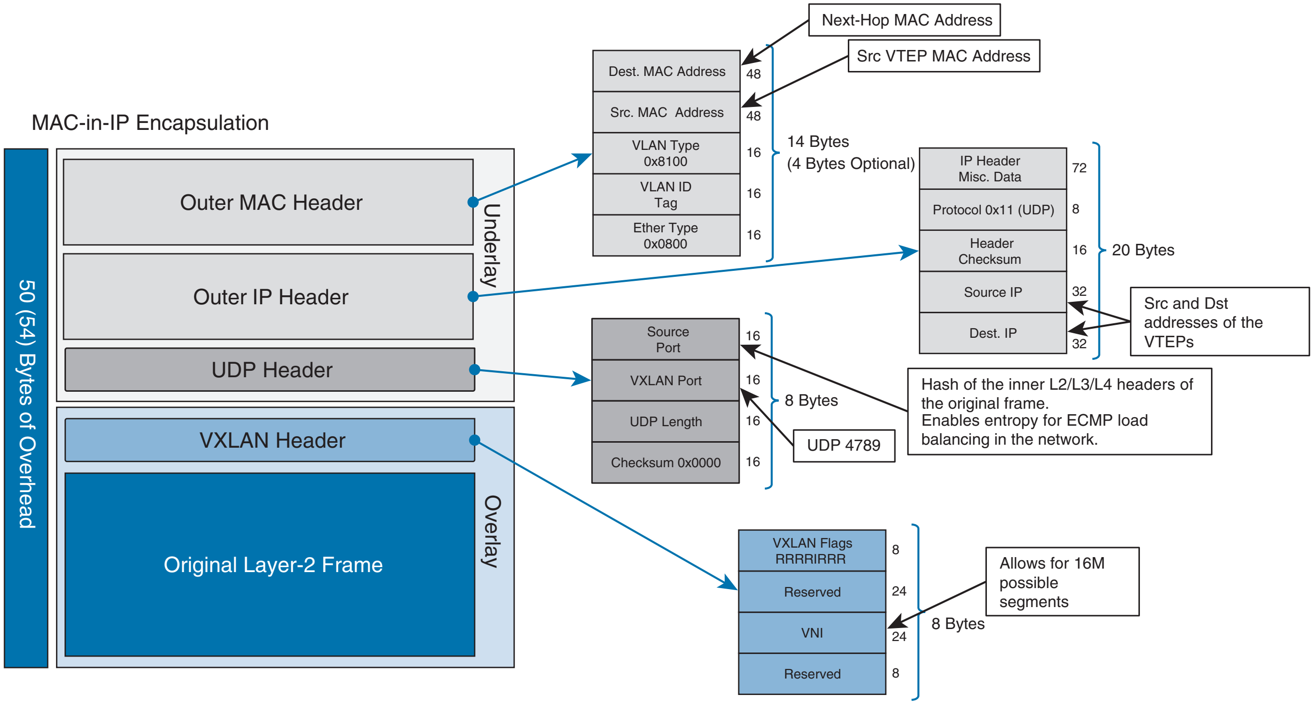


Figure 3-6 *VXLAN Frame Format*

Micro-Segmentation

Key Topic

For decades, servers were assigned subnets and VLANs. Sounds pretty simple, right? Well, this introduced a lot of complexities because application segmentation and policies were physically restricted to the boundaries of the VLAN within the same data center (or even in “the campus”). In virtual environments, the problem became harder. Nowadays applications can move around between servers to balance loads for performance or high availability upon failures. They also can move between different data centers and even different cloud environments.

Traditional segmentation based on VLANs constrains you to maintain the policies of which application needs to talk to which application (and who can access such applications) in centralized firewalls. This is ineffective because most traffic in data centers is now “East-West” traffic. A lot of that traffic does not even hit the traditional firewall. In virtual environments, a lot of the traffic does not even leave the physical server.

Key Topic

Let’s define what people refer to as “East-West” traffic and “North-South” traffic. “East-West” traffic is network traffic between servers (virtual servers or physical servers, containers, and so on).

“North-South” traffic is network traffic flowing in and outside the data center. Figure 3-7 illustrates the concepts of “East-West” and “North-South” traffic.

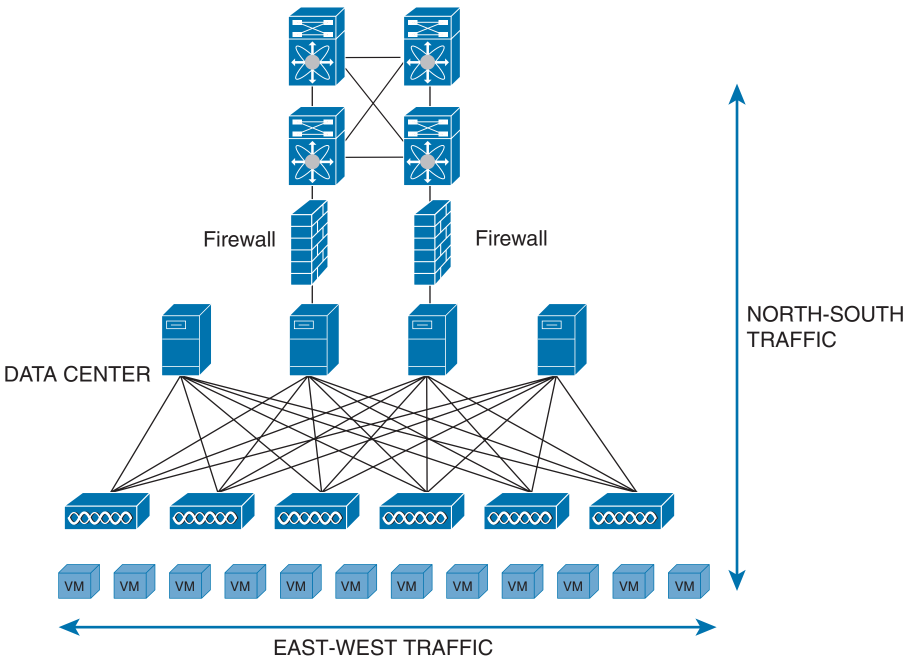


Figure 3-7 *“East-West” and “North-South” Traffic*

Many vendors have created solutions where policies applied to applications are independent from the location or the network tied to the application.

For example, let’s suppose that you have different applications running in separate VMs and those applications also need to talk to a database (as shown in Figure 3-8).

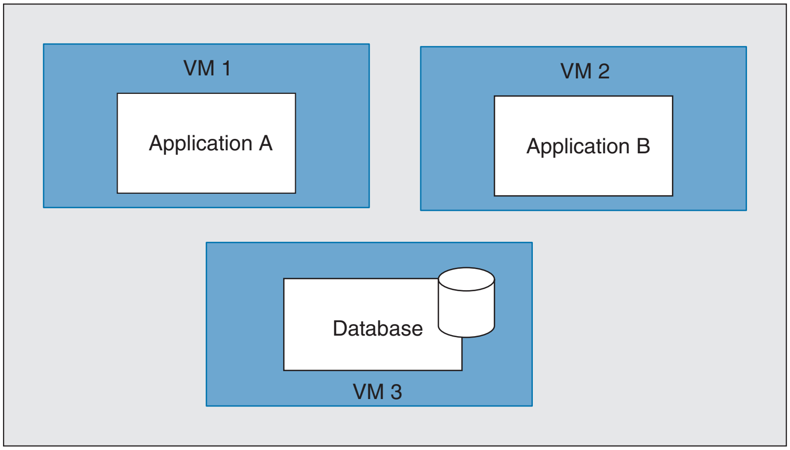


Figure 3-8 *Applications in VMs*

You need to apply policies to restrict if application A needs or does not need to talk to application B, or which application should be able to talk to the database. These policies should not be bound by which VLAN or IP subnet the application belongs to and whether it is in the same rack or even in the same data center. Network traffic should not make multiple trips back and forth between the applications and centralized firewalls to enforce policies between VMs.

Containers make this a little harder because they move and change more often. Figure 3-9 illustrates a high-level representation of applications running inside of containers (for example, Docker containers).

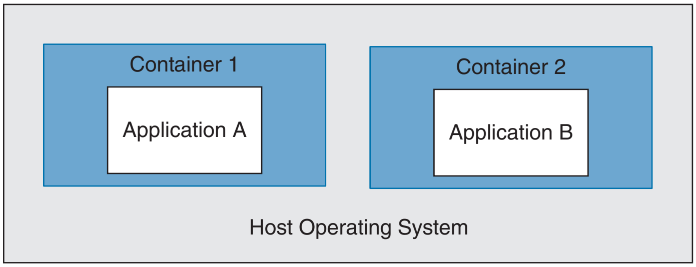


Figure 3-9 *Applications in Containers*

The ability to enforce network segmentation in those environments is what's called "micro-segmentation." Micro-segmentation is at the VM level or between containers regardless of a VLAN or a subnet. Micro-segmentation segmentation solutions need to be "application aware." This means that the segmentation process starts and ends with the application itself.

Most micro-segmentation environments apply a "zero-trust model." This model dictates that users cannot talk to applications, and applications cannot talk to other applications unless a defined set of policies permits them to do so.

Key Topic

Open Source Initiatives

There are several open source projects that are trying to provide micro-segmentation and other modern networking benefits. Examples include the following:

- Neutron from OpenStack
- Open vSwitch (OVS)
- Open Virtual Network (OVN)
- OpenDaylight (ODL)
- Open Platform for Network Function Virtualization (OPNFV)
- Contiv

The concept of SDN is very broad, and every open source provider and commercial vendor takes it in a different direction. The networking component of OpenStack is called Neutron. Neutron is designed to provide "networking as a service" in private, public, and hybrid cloud environments. Other OpenStack components, such as Horizon (Web UI) and Nova (compute service), interact with Neutron using a set of APIs to configure the networking services. Neutron uses plug-ins to deliver advanced networking capabilities and allow third-party vendor integration. Neutron has two main components: the neutron server and a database that handles persistent storage and plug-ins to provide additional services. Additional information about Neutron and OpenStack can be found at https://docs.openstack.org/neutron/latest.

OVN was originally created by the folks behind Open vSwitch (OVS) for the purpose of bringing an open source solution for virtual network environments and SDN. Open vSwitch is an open source implementation of a multilayer virtual switch inside the hypervisor.

NOTE You can download Open vSwitch and access its documentation at https://www.openvswitch.org.

OVN is often used in OpenStack implementations with the use of OVS. You can also use OVN with the OpenFlow protocol. OpenStack Neutron uses OVS as the default "control plane."

NOTE You can access different tutorials about OVN and OVS at http://docs.openvswitch.org/en/latest/tutorials/.

OpenDaylight (ODL) is another popular open source project that is focused on the enhancement of SDN controllers to provide network services across multiple vendors. OpenDaylight participants also interact with the OpenStack Neutron project and attempt to solve the existing inefficiencies.

OpenDaylight interacts with Neutron via a northbound interface and manages multiple interfaces southbound, including the Open vSwitch Database Management Protocol (OVSDB) and OpenFlow.

TIP You can find more information about OpenDaylight at https://www.opendaylight.org. Cisco has several tutorials and additional information about OpenDaylight in DevNet at https://developer.cisco.com/site/opendaylight/.

Key Topic

So, what is a northbound and southbound API? In an SDN architecture, southbound APIs are used to communicate between the SDN controller and the switches and routers within the infrastructure. These APIs can be open or proprietary.

NOTE Cisco provides detailed information about the APIs supported in all platforms in DevNet (developer.cisco.com). DevNet will be discussed in detail later in this chapter.

Southbound APIs enable SDN controllers to dynamically make changes based on real-time demands and scalability needs. OpenFlow and Cisco OpFlex provide southbound API capabilities.

Northbound APIs (SDN northbound APIs) are typically RESTful APIs that are used to communicate between the SDN controller and the services and applications running over the network. Such northbound APIs can be used for the orchestration and automation of the network components to align with the needs of different applications via SDN network programmability. In short, northbound APIs are basically the link between the applications and the SDN controller. In modern environments, applications can tell the network devices (physical or virtual) what type of resources they need and, in turn, the SDN solution can provide the necessary resources to the application.

Cisco has the concept of intent-based networking. On different occasions, you may see northbound APIs referred to as "intent-based APIs."

Key Topic

More About Network Function Virtualization

Network virtualization is used for logical groupings of nodes on a network. The nodes are abstracted from their physical locations so that VMs and any other assets can be managed as if they are all on the same physical segment of the network. This is not a new technology. However, it is still one that is key in virtual environments where systems are created and moved despite their physical location.

Network Functions Virtualization (NFV) is a technology that addresses the virtualization of Layer 4 through Layer 7 services. These include load balancing and security capabilities such as firewall-related features. In short, with NFV, you convert certain types of network appliances into VMs. NFV was created to address the inefficiencies that were introduced by virtualization.

NFV allows you to create a virtual instance of a virtual node such as a firewall that can be deployed where it is needed, in a flexible way that's similar to how you do with a traditional VM.

Open Platform for Network Function Virtualization (OPNFV) is an open source solution for NFV services. It aims to be the base infrastructure layer for running virtual network functions. You can find detailed information about OPNFV at opnfv.org.

NFV nodes such as virtual routers and firewalls need an underlying infrastructure:

- A hypervisor to separate the virtual routers, switches, and firewalls from the underlying physical hardware. The hypervisor is the underlying virtualization platform that allows the physical server (system) to operate multiple VMs (including traditional VMs and network-based VMs).
- A virtual forwarder to connect individual instances.
- A network controller to control all of the virtual forwarders in the physical network.
- A VM manager to manage the different network-based VMs.

Figure 3-10 demonstrates the high-level components of the NFV architecture.

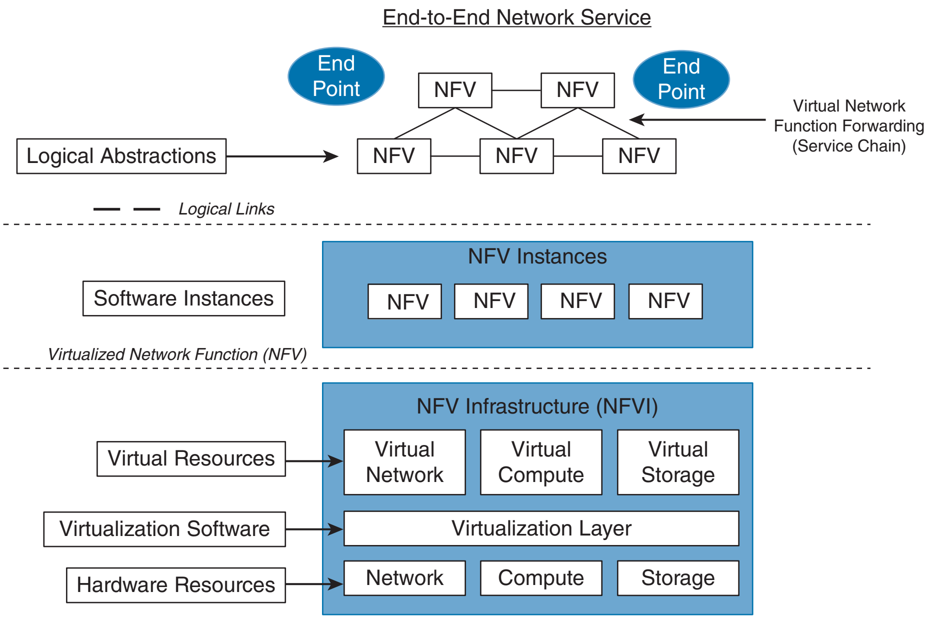


Figure 3-10 *NFV Architecture*

Several NFV infrastructure components have been created in open community efforts. On the other hand, traditionally, the actual integration has so far remained a "private" task. You've either had to do it yourself, outsource it, or buy a pre-integrated system from some vendor, keeping in mind that the systems integration undertaken is not a one-time task. OPNFV was created to change the NFV ongoing integration task from a private solution into an open community solution.

NFV MANO

NFV changes the way networks are managed. NFV management and network orchestration (MANO) is a framework and working group within the European Telecommunications Standards Institute (ETSI) Industry Specification Group for NFV (ETSI ISG NFV). NFV MANO is designed to provide flexible on-boarding of network components. NFV MANO is divided into the three functional components listed in Figure 3-11.

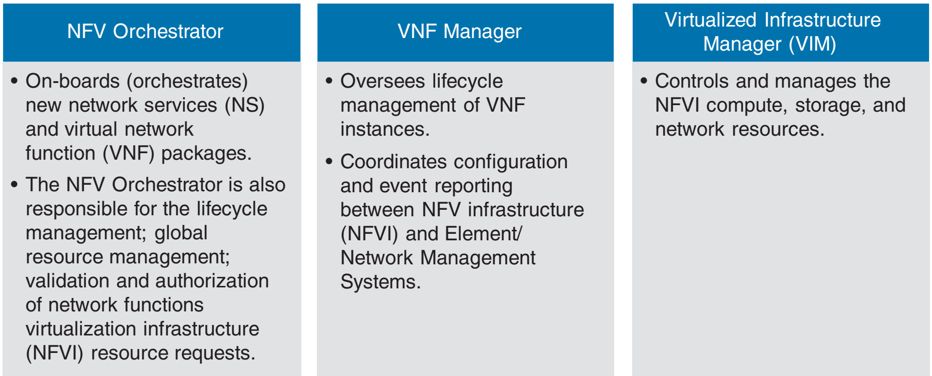


Figure 3-11 *NFV MANO Functional Components*

The NFV MANO architecture is integrated with open application program interfaces (APIs) in the existing systems. The MANO layer works with templates for standard VNFs. It allows implementers to pick and choose from existing NFV resources to deploy their platform or element.

Contiv

Contiv is an open source project that allows you to deploy micro-segmentation policy-based services in container environments. It offers a higher level of networking abstraction for microservices by providing a policy framework. Contiv has built-in service discovery and service routing functions to allow you to scale out services.

NOTE You can download Contiv and access its documentation at https://contiv.io.

With Contiv you can assign an IP address to each container. This feature eliminates the need for host-based port NAT. Contiv can operate in different network environments such as traditional Layer 2 and Layer 3 networks, as well as overlay networks.

Contiv can be deployed with all major container orchestration platforms (or schedulers) such as Kubernetes and Docker Swarm. For instance, Kubernetes can provide compute resources to containers and then Contiv provides networking capabilities.

NOTE Contiv supports Layer 2, Layer 3 (BGP), VXLAN for overlay networks, and Cisco ACI mode. It also provides built-in east-west service load balancing and traffic isolation.

The Netmaster and Netplugin (Contiv host agent) are the two major components in Contiv. Figure 3-12 illustrates how the Netmaster and the Netplugin interact with all the underlying components of the Contiv solution.

TIP The Contiv website includes several tutorials and step-by-step integration documentation at https://contiv.io/documents/tutorials/index.html.

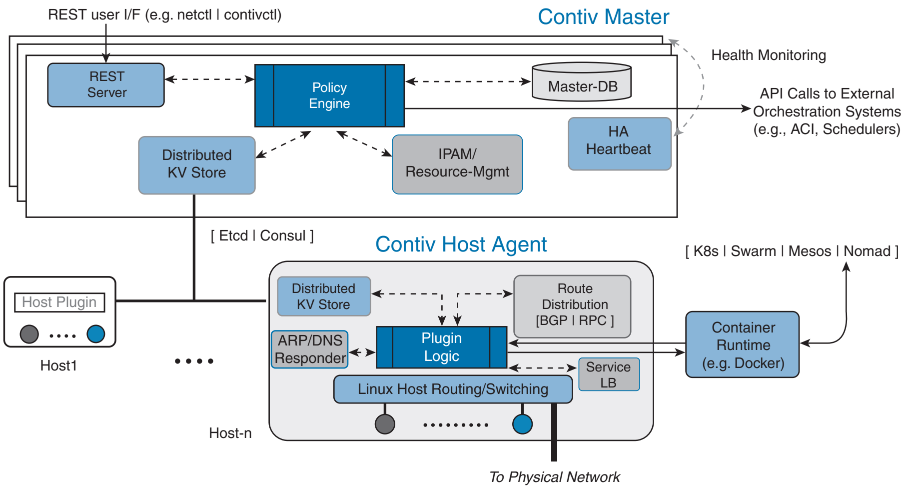


Figure 3-12 *Contiv Netmaster and Netplugin (Contiv Host Agent) Components*

Cisco Digital Network Architecture (DNA)

Cisco DNA is a solution created by Cisco that is often referred to as the "intent-based networking" solution. Cisco DNA provides automation and assurance services across campus networks, wide area networks (WANs), and branch networks. Cisco DNA is based on an open and extensible platform and provides the policy, automation, and analytics capabilities, as illustrated in Figure 3-13.

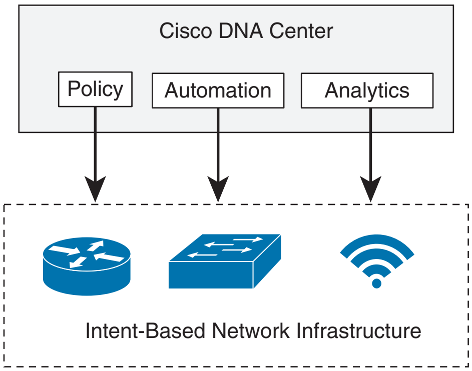


Figure 3-13 *Cisco DNA High-Level Architecture*

The heart of the Cisco DNA solution is Cisco DNA Center (DNAC). DNAC is a command-and-control element that provides centralized management via dashboards and APIs. Figure 3-14 shows one of the many dashboards of Cisco DNA Center (the Network Hierarchy dashboard).

Cisco DNA Center can be integrated with external network and security services such as the Cisco Identity Services Engine (ISE). Figure 3-15 shows how the Cisco ISE is configured as an authentication, authorization, and accounting (AAA) server in the Cisco DNA Center Network Settings screen.

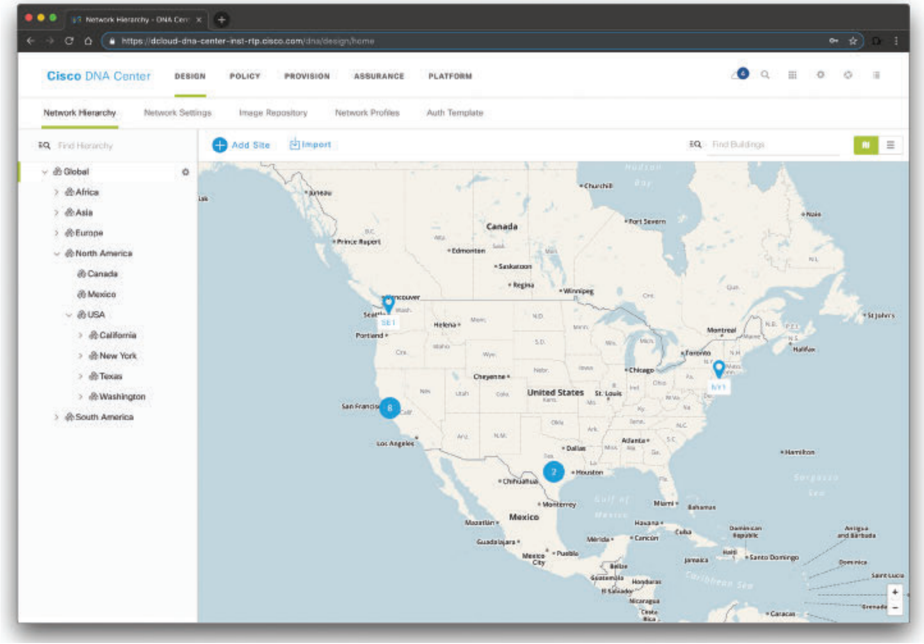


Figure 3-14 *Cisco DNA Center Network Hierarchy Dashboard*

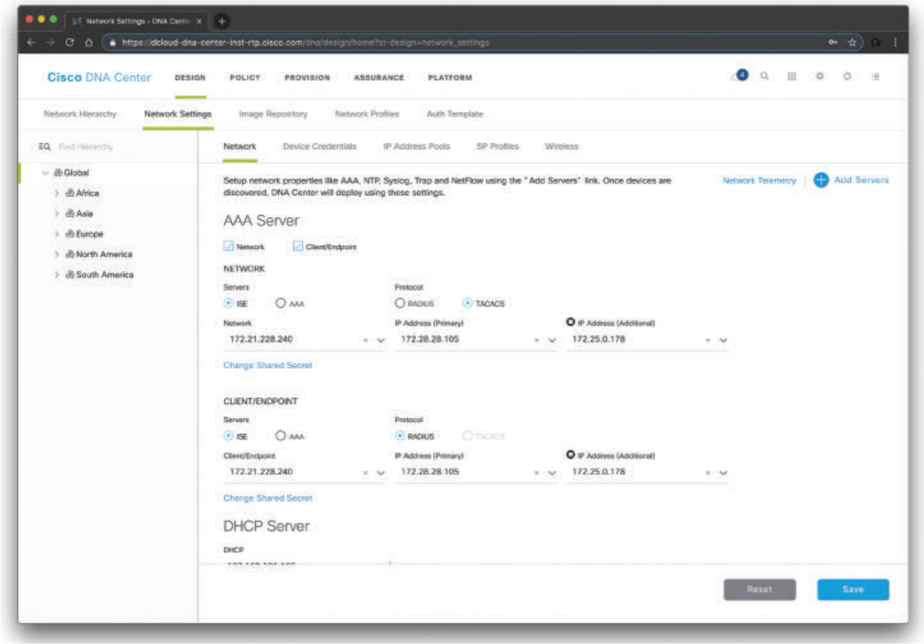


Figure 3-15 *Cisco DNA Center Integration with Cisco ISE for AAA Services*

Cisco DNA Policies

The following are the policies you can create in the Cisco DNA Center:

- Group-based access control policies
- IP-based access control policies
- Application access control policies
- Traffic copy policies

Figure 3-16 shows the Cisco DNA Center Policy Dashboard. There you can see the number of virtual networks, group-based access control policies, IP-based access control policies, traffic copy policies, scalable groups, and IP network groups that have been created. The Policy Dashboard will also show any policies that have failed to deploy.

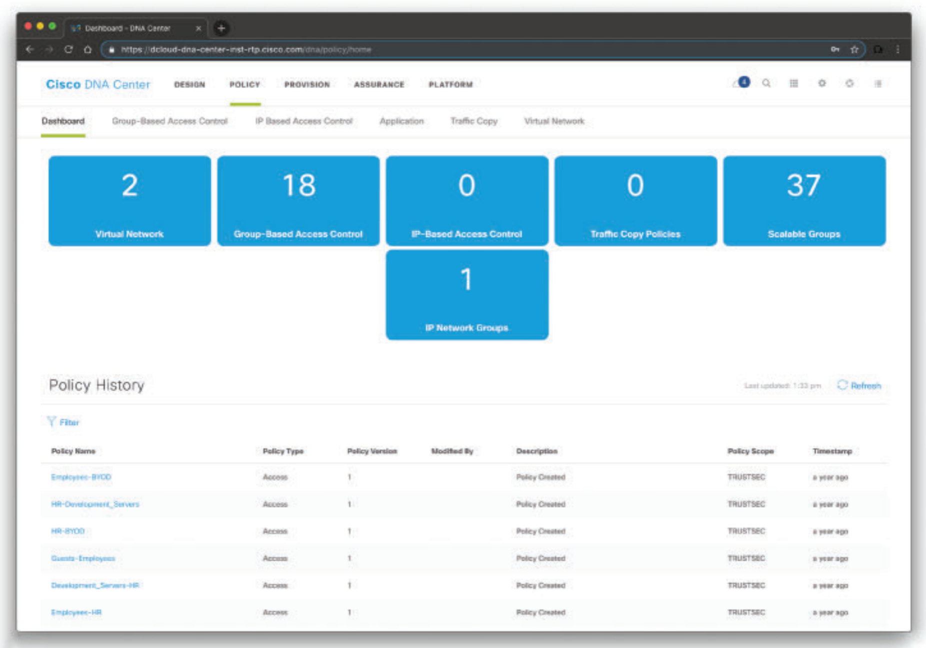


Figure 3-16 *Cisco DNA Center Policy Dashboard*

The Policy Dashboard window also provides a list of policies and the following information about each policy:

- **Policy Name:** The name of the policy.
- **Policy Type:** The type of policy.
- **Policy Version:** The version number is incremented by one version each time you change a policy.

- **Modified By:** The user who created or modified the policy.
- **Description:** The policy description.
- **Policy Scope:** The policy scope defines the users and device groups or applications that a policy affects.
- **Timestamp:** The date and time when a particular version of a policy was saved.

Cisco DNA Group-Based Access Control Policy

When you configure group-based access control policies, you need to integrate the Cisco ISE with Cisco DNA Center, as you learned previously in this chapter. In Cisco ISE, you configure the work process setting as "Single Matrix" so that there is only one policy matrix for all devices in the TrustSec network. You will learn more about Cisco TrustSec and Cisco ISE in Chapter 4, "Authentication, Authorization, Accounting (AAA) and Identity Management."

Depending on your organization's environment and access requirements, you can segregate your groups into different virtual networks to provide further segmentation.

After Cisco ISE is integrated in Cisco DNA Center, the scalable groups that exist in Cisco ISE are propagated to Cisco DNA Center. If a scalable group that you need does not exist, you can create it in Cisco ISE.

> **NOTE** You can access Cisco ISE through the Cisco DNA Center interface to create scalable groups. After you have added a scalable group in Cisco ISE, it is synchronized with the Cisco DNA Center database so that you can use it in an access control policy. You cannot edit or delete scalable groups from Cisco DNA Center; you need to perform these tasks from Cisco ISE.

Cisco DNA Center has the concept of access control contracts. A contract specifies a set of rules that allow or deny network traffic based on such traffic matching particular protocols or ports. Figure 3-17 shows a new contract being created in Cisco DNA Center to allow SSH access (TCP port 22).

To create a contract, navigate to **Policy > Group-Based Access Control > Access Contract** and click **Add Contract**. The dialog box shown in Figure 3-17 will be displayed.

Figure 3-18 shows an example of how to create a group-based access control policy.

In Figure 3-18, an access control policy named **omar_policy_1** is configured to **deny** traffic from all users and related devices in the group called **Guests** to any user or device in the **Finance** group.

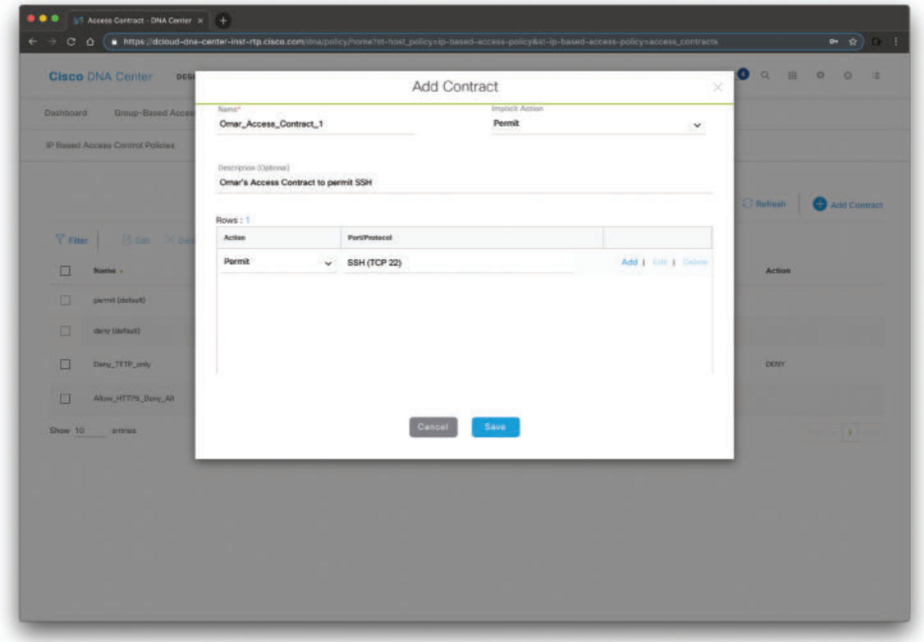


Figure 3-17 *Adding a Cisco DNA Center Contract*

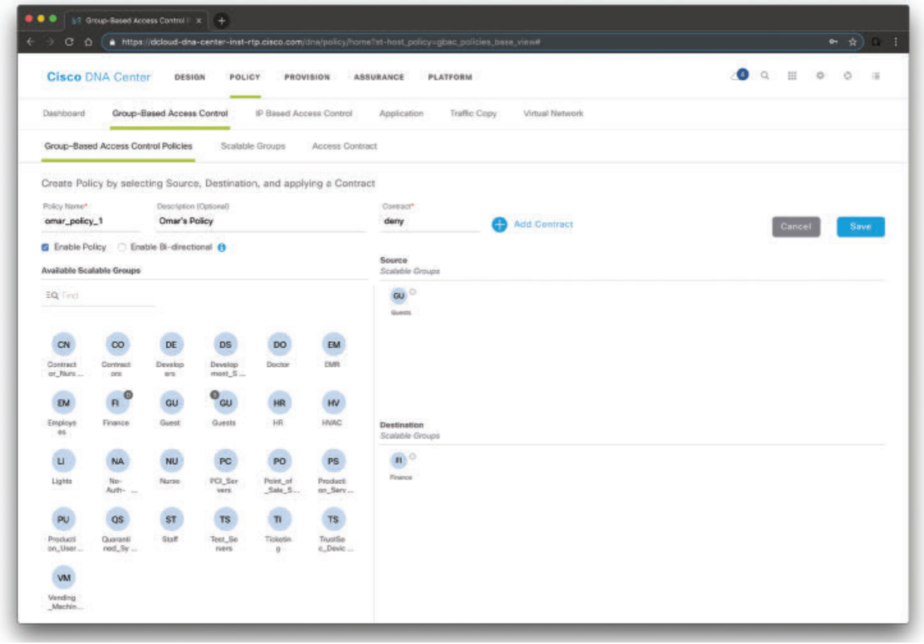


Figure 3-18 *Adding a Cisco DNA Center Group-Based Access Control Policy*

Cisco DNA IP-Based Access Control Policy

You can also create IP-based access control policies in Cisco DNA Center. To create IP-based access control policies, navigate to **Policy > IP Based Access Control > IP Based Access Control Policies**, as shown in Figure 3-19.

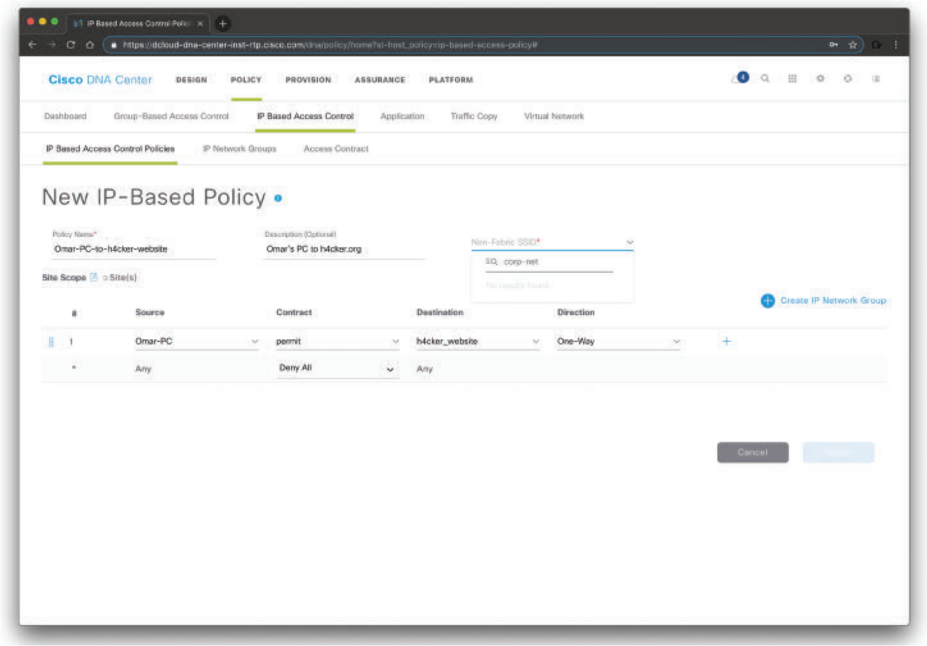


Figure 3-19 *Adding a Cisco DNA Center IP-Based Access Control Policy*

In the example shown in Figure 3-19, a policy is configured to permit Omar's PC to communicate with h4cker.org.

NOTE An IP network group named h4cker_website is already configured. To configure IP network groups, navigate to **Policy > IP Based Access Control > IP Network Groups**. These IP network groups can also be automatically populated from Cisco ISE.

You can also associate these policies to specific wireless SSIDs. The **corp-net** SSID is associated to the policy entry in Figure 3-19.

Cisco DNA Application Policies

Application policies can be configured in Cisco DNA Center to provide Quality of Service (QoS) capabilities. The following are the Application Policy components you can configure in Cisco DNA Center:

- Applications
- Application sets

- Application policies
- Queuing profiles

Applications in Cisco DNA Center are the software programs or network signaling protocols that are being used in your network.

NOTE Cisco DNA Center supports all of the applications in the Cisco Next Generation Network-Based Application Recognition (NBAR2) library.

Applications can be grouped into logical groups called *application sets*. These application sets can be assigned a business relevance within a policy.

You can also map applications to industry standard-based traffic classes, as defined in RFC 4594.

Cisco DNA Traffic Copy Policy

You can also use an Encapsulated Remote Switched Port Analyzer (ERSPAN) configuration in Cisco DNA Center so that the IP traffic flow between two entities is copied to a given destination for monitoring or troubleshooting. In order for you to configure ERSPAN using Cisco DNA Center, you need to create a traffic copy policy that defines the source and destination of the traffic flow you want to copy. To configure a traffic copy policy, navigate to **Policy > Traffic Copy > Traffic Copy Policies**, as shown in Figure 3-20.

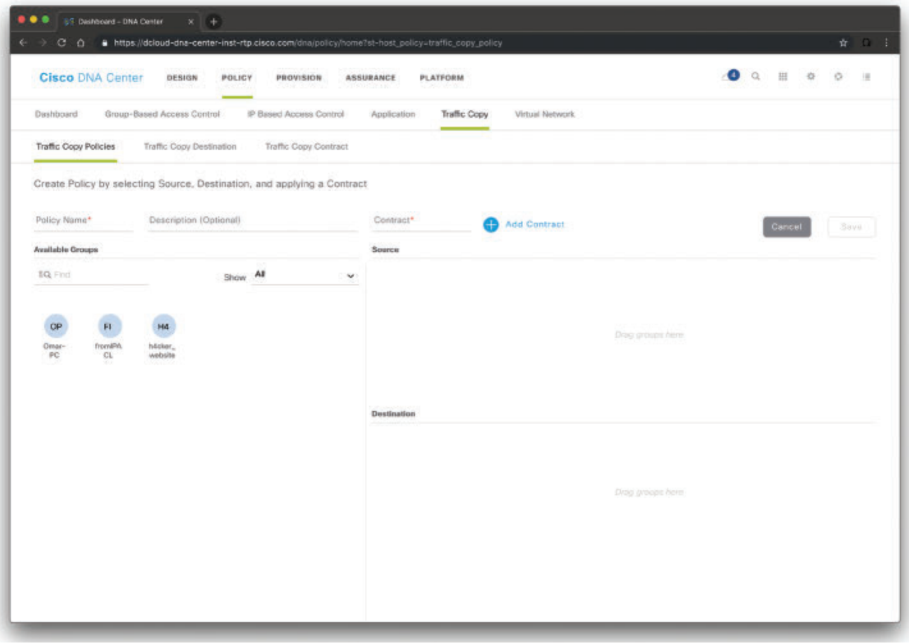


Figure 3-20 *Adding a Traffic Copy Policy*

You can also define a traffic copy contract that specifies the device and interface where the copy of the traffic is sent.

Cisco DNA Center Assurance Solution

The Cisco DNA Center Assurance solution allows you to get contextual visibility into network functions with historical, real-time, and predictive insights across users, devices, applications, and the network. The goal is to provide automation capabilities to reduce the time spent on network troubleshooting.

Figure 3-21 shows the Cisco DNA Center Assurance Overall Health dashboard.

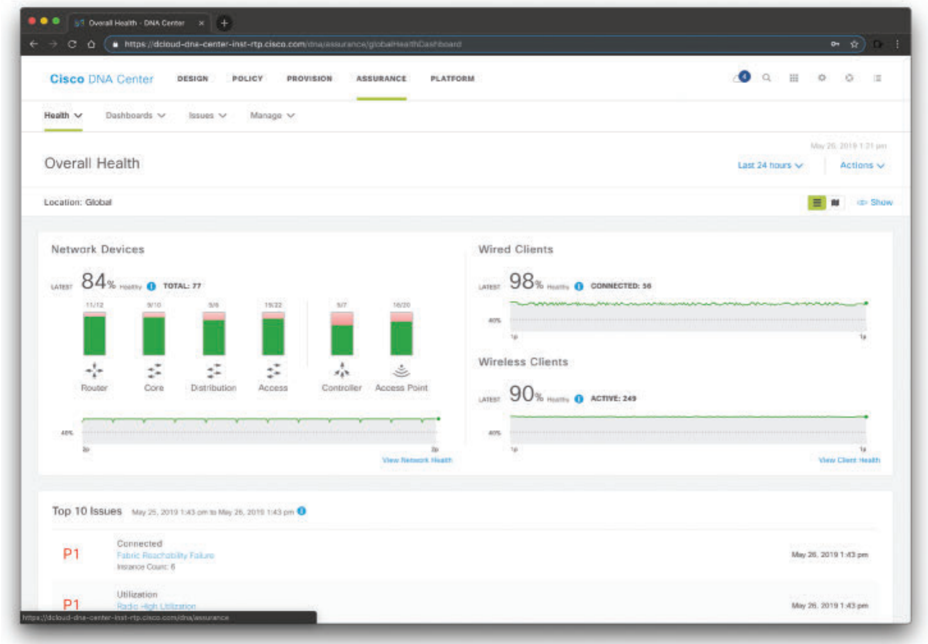


Figure 3-21 *The Cisco DNA Center Assurance Overall Health Dashboard*

The Cisco DNA Center Assurance solution allows you to investigate different networkwide (global) issues, as shown in Figure 3-22.

The Cisco DNA Center Assurance solution also allows you to configure sensors to test the health of wireless networks. A wireless network includes access point (AP) radios, WLAN configurations, and wireless network services. Sensors can be dedicated or on-demand sensors. A dedicated sensor is when an AP is converted into a sensor, and it stays in sensor mode (is not used by wireless clients) unless it is manually converted back into AP mode. An on-demand sensor is when an AP is temporarily converted into a sensor to run tests. After the tests are complete, the sensor goes back to AP mode. Figure 3-23 shows the Wireless Sensor dashboard in Cisco DNA Center.

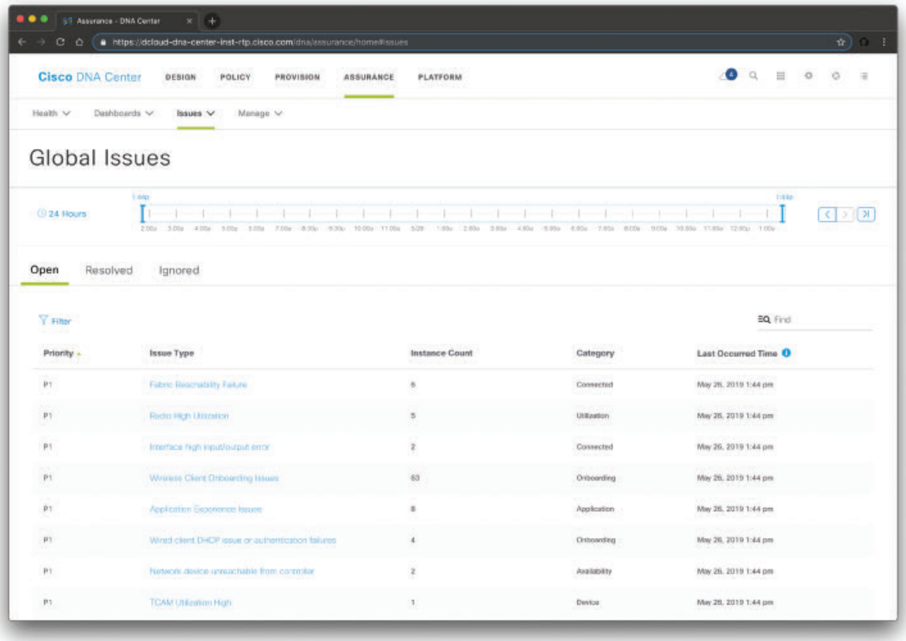


Figure 3-22 *The Cisco DNA Center Assurance Global Issues Dashboard*

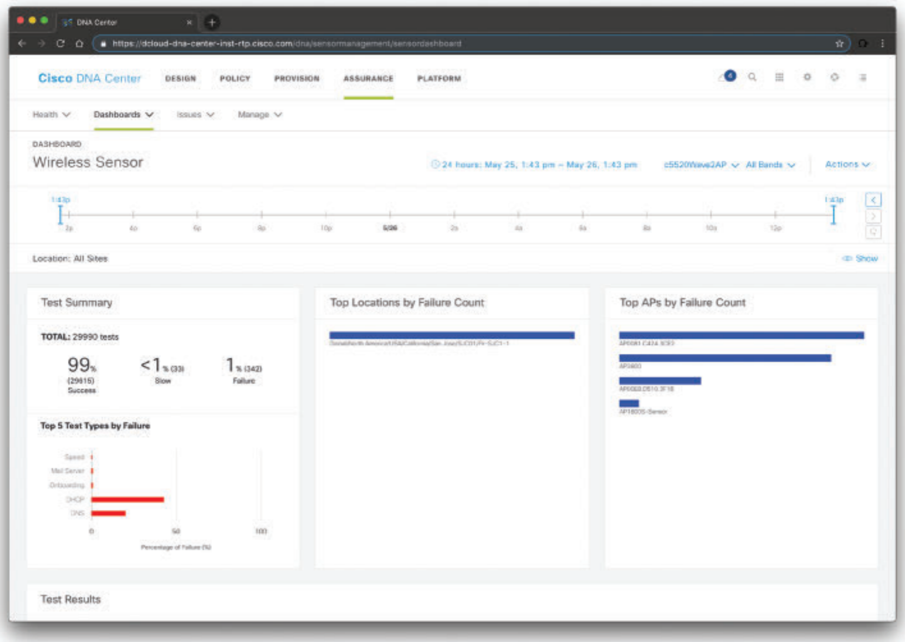


Figure 3-23 *The Cisco DNA Center Assurance Wireless Sensor Dashboard*

Key Topic

Cisco DNA Center APIs

One of the key benefits of the Cisco DNA Center is the comprehensive available APIs (aka Intent APIs). The Intent APIs are northbound REST APIs that expose specific capabilities of the Cisco DNA Center platform. These APIs provide policy-based abstraction of business intent, allowing you to focus on an outcome to achieve instead of struggling with the mechanisms that implement that outcome. The APIs conform to the REST API architectural style and are simple, extensible, and secure to use.

Cisco DNA Center also has several integration APIs. These integration capabilities are part of westbound interfaces. Cisco DNA Center also allows administrators to manage their non-Cisco devices. Multivendor support comes to Cisco DNA Center through the use of an SDK that can be used to create device packages for third-party devices. A device package enables Cisco DNA Center to communicate with third-party devices by mapping Cisco DNA Center features to their southbound protocols.

TIP Cisco has very comprehensive documentation and tutorials about the Cisco DNA Center APIs at DevNet (https://developer.cisco.com/dnacenter).

Cisco DNA Center also has several events and notifications services that allow you to capture and forward Cisco DNA Assurance and Automation (SWIM) events to third-party applications via a webhook URL.

All Cisco DNA Center APIs conform to the REST API architectural styles.

NOTE A REST endpoint accepts and returns HTTPS messages that contain JavaScript Object Notation (JSON) documents. You can use any programming language to generate the messages and the JSON documents that contain the API methods. These APIs are governed by the Cisco DNA Center Role-Based Access Control (RBAC) rules and as a security measure require the user to authenticate successfully prior to using the API.

You can view information about all the Cisco DNA Center APIs by clicking the **Platform** tab and navigating to **Developer Toolkit > APIs**, as shown in Figure 3-24.

Figure 3-25 shows an example of the detailed API documentation within Cisco DNA Center.

Key Topic

TIP All REST requests in Cisco DNA Center require authentication. The Authentication API generates a security token that encapsulates the privileges of an authenticated REST caller. All requested operations are authorized by Cisco DNA Center according to the access privileges associated with the security token that is sent in the request.

Cisco is always expanding the capabilities of the Cisco DNA Center APIs. Please study and refer to the following API documentation and tutorials for the most up-to-date capabilities: https://developer.cisco.com/docs/dna-center and https://developer.cisco.com/site/dna-center-rest-api.

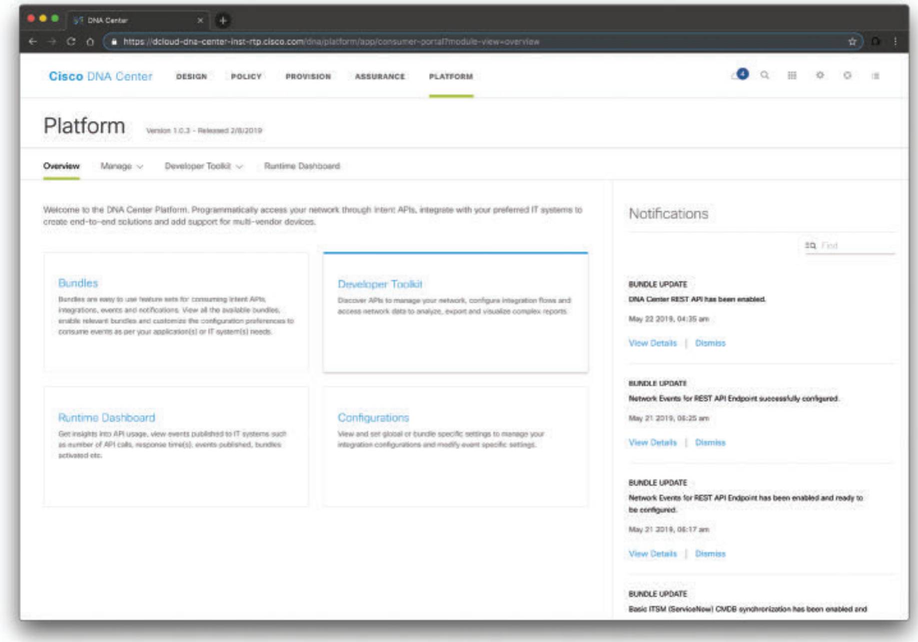


Figure 3-24 *The Cisco DNA Center APIs and Developer Toolkit*

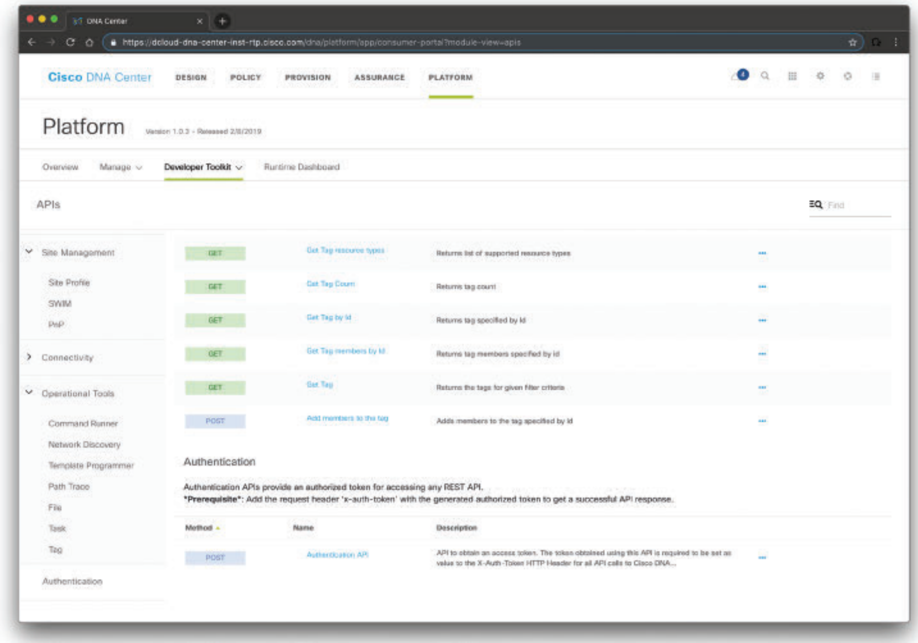


Figure 3-25 *API Developer Toolkit Documentation*

Key Topic

Cisco DNA Security Solution

The Cisco DNA Security solution supports several other security products and operations that allow you to detect and contain cybersecurity threats. One of the components of the Cisco DNA Security solution is the Encrypted Traffic Analytics (ETA) solution. Cisco ETA allows you to detect security threats in encrypted traffic without decrypting the packets. It is able to do this by using machine learning and other capabilities. To use Encrypted Traffic Analytics, you need one of the following network devices along with Cisco Stealthwatch Enterprise:

- Catalyst 9000 switches
- ASR 1000 Series routers
- ISR 4000 Series routers
- CSR 1000V Series virtual routers
- ISR 1000 Series routers
- Catalyst 9800 Series wireless controllers

Cisco Stealthwatch provides network visibility and security analytics to rapidly detect and contain threats. You will learn more about the Cisco Stealthwatch solution in Chapter 5, "Network Visibility and Segmentation."

As you learned in previous sections of this chapter, the Cisco TrustSec solution and Cisco ISE enable you to control networkwide access, enforce security policies, and help meet compliance requirements.

Cisco DNA Multivendor Support

Cisco DNA Center now allows customers to manage their non-Cisco devices. Multivendor support comes to Cisco DNA Center through the use of an SDK that can be used to create device packages for third-party devices. A device package enables Cisco DNA Center to communicate with third-party devices by mapping Cisco DNA Center features to their southbound protocols. Multivendor support capabilities are based on southbound interfaces. These interfaces interact directly with network devices by means of CLI, SNMP, or NETCONF.

NOTE Southbound interfaces are not exposed to the consumer. Instead, the consumer uses Intent APIs, which abstract the underlying complexity of the traditional network. The user of Intent APIs need not be concerned with the particular protocols that the southbound interfaces use to implement network intent on devices that Cisco DNA Center supports.

Introduction to Network Programmability

As you were able to see in previous sections of this chapter, learning to code and work with programmable infrastructures is very important in today's environment. You saw the value of using APIs. Whether you have configured large networks in the past or are just getting started, you know that this probably involved a lot of clicking, typing, copying-and-pasting, and many repetitive tasks. Nowadays, modern APIs enable you to complete powerful tasks, reduce all the repetitive work, and save time.

Using APIs, you can make requests like the ones shown in Figure 3-26 in a very simple way.

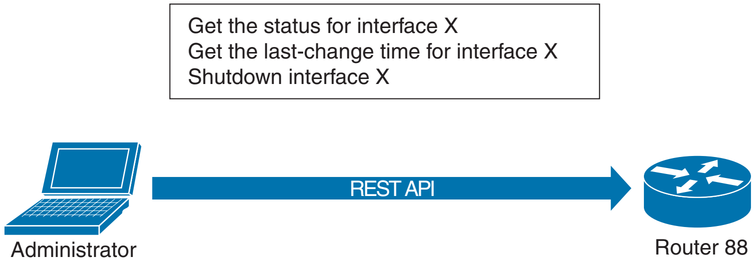


Figure 3-26 *Using Network Infrastructure Device APIs*

3

Key Topic

Modern Programming Languages and Tools

Modern programming languages like JavaScript, Python, Go, Swift, and others are more flexible and easier to learn than their predecessors. You might wonder what programming language you should learn first. Python is one of the programming languages recommended to learn first—not only for network programmability, but for many other scenarios.

> **TIP** Many different sites allow you to get started with Python. The following are several great resources to learn Python:
>
> - Learn Python dot org: https://www.learnpython.org
> - W3 Schools Python tutorials: https://www.w3schools.com/python/
> - The Python Tutorial: https://docs.python.org/3/tutorial/

Combining programming capabilities with developer tools like Git (GitHub or GitLab repositories), package management systems, virtual environments, and integrated development environments (IDEs) allows you to create your own set of powerful tools and workflows.

Another amazing thing is the power of code reuse and online communities. In the past, when you wanted to create some program, you often had to start "from scratch." For example, if you wanted to just make an HTTPS web request, you had to create code to open a TCP connection over port 443, perform the TLS negotiation, exchange and validate certificates, and format and interpret HTTP requests and responses.

Nowadays, you can just use open source software in GitHub or simply use packages such as the Python requests package, as shown in Figure 3-27.

In Figure 3-27, the Python package called *requests* is installed using the package manager for Python called *pip* (https://pypi.org/project/pip). The requests library allows you to make HTTP/HTTPS requests in Python very easily.

Now that you have the requests package installed, you can start making HTTP requests, as shown in Figure 3-28.

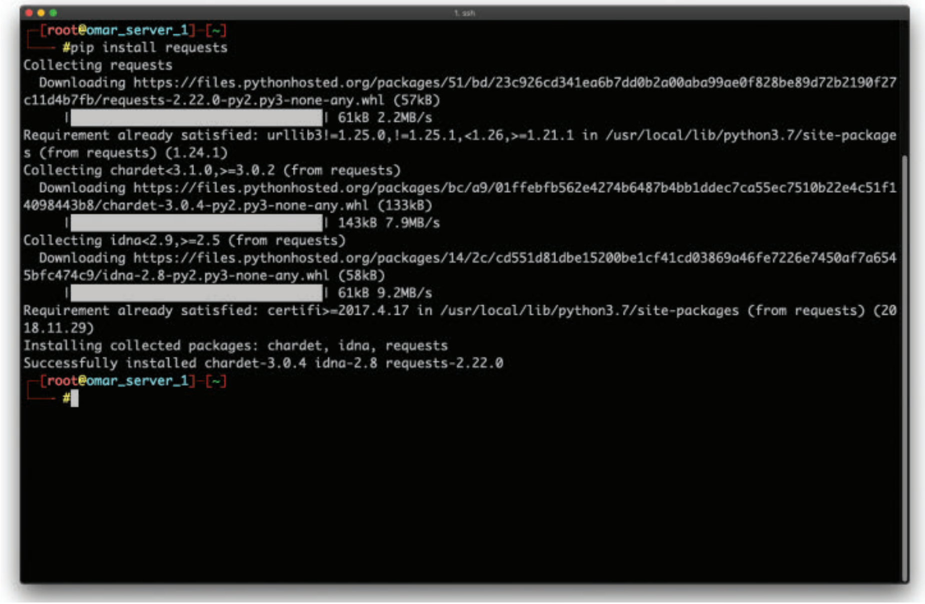


Figure 3-27 *Installing the Python Requests Package Using pip*

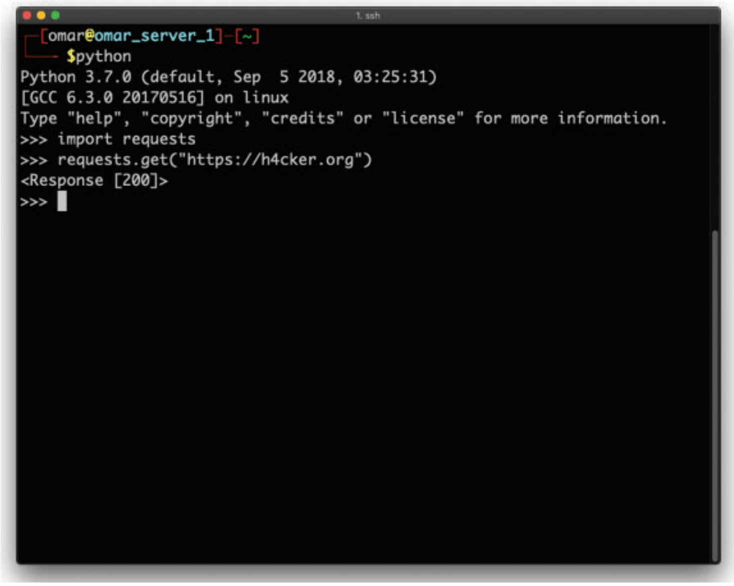


Figure 3-28 *Using the Python Requests Package*

In Figure 3-28, the interactive Python shell (interpreter) is used to use (import) the requests package and send an HTTP GET request to the website at https://h4cker.org. The HTTP GET request is successful and the 200 message/response is shown.

Additional information about the Python interpreter can be found at https://docs.python.org/3/tutorial/interpreter.html and https://www.python-course.eu/python3_interactive.php.

TIP The W3 schools website has a very good explanation of the HTTP status code messages at https://www.w3schools.com/tags/ref_httpmessages.asp.

The HTTP status code messages can be in the following ranges:

- Messages in the 100 range are informational.
- Messages in the 200 range are related to successful transactions.
- Messages in the 300 range are related to HTTP redirections.
- Messages in the 400 range are related to client errors.
- Messages in the 500 range are related to server errors.

When HTTP servers and browsers communicate with each other, they perform interactions based on headers as well as body content. The HTTP Request has the following structure:

1. The METHOD, which in this example is an HTTP GET. However, the HTTP methods can be the following:
   - **GET:** Retrieves information from the server.
   - **HEAD:** Basically, this is the same as a GET, but it returns only HTTP headers and no document body.
   - **POST:** Sends data to the server (typically using HTML forms, API requests, and the like).
   - **TRACE:** Does a message loopback test along the path to the target resource.
   - **PUT:** Uploads a representation of the specified URI.
   - **DELETE:** Deletes the specified resource.
   - **OPTIONS:** Returns the HTTP methods that the server supports.
   - **CONNECT:** Converts the request connection to a transparent TCP/IP tunnel.
2. The URI and the path-to-resource field represent the path portion of the requested URL.
3. The request version-number field specifies the version of HTTP used by the client.
4. The user agent is Chrome in this example, and it was used to access the website. In the packet capture, you see the following:

   ```
   User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181
   Safari/537.36\r\n.
   ```
5. Next, you see several other fields like accept, accept-language, accept encoding, and others.
6. The server, after receiving this request, generates a response.
7. The server response has a three-digit status code and a brief human-readable explanation of the status code. Then below you see the text data (which is the HTML code coming back from the server and displaying the website contents).

TIP The requests Python package is used often to interact with APIs. You can obtain more information about the requests Python package at https://realpython.com/python-requests and https://developer.cisco.com/learning/lab/intro-python-part1/step/1.

Key Topic

DevNet

DevNet is a platform created by Cisco that has numerous resources for network and application developers. DevNet is an amazing resource that includes many tutorials, free video courses, sandboxes, learning paths, and sample code to interact with many APIs. You can access DevNet at developer.cisco.com.

If you are new to programming and network programmability, you can take advantage of the following DevNet tutorials and learning paths:

- Introduction to Coding and APIs: https://developer.cisco.com/startnow
- Network Programmability Basics Video Course: https://developer.cisco.com/video/net-prog-basics/
- Parsing JSON using Python: https://developer.cisco.com/learning/lab/coding-202-parsing-json/step/1
- DevNet GitHub Repositories: https://github.com/CiscoDevNet
- DevNet Developer Videos: https://developer.cisco.com/video
- DevNet Git Tutorials: https://developer.cisco.com/learning/lab/git-intro/step/1
- DevNet ACI Programmability: https://developer.cisco.com/learning/tracks/aci-programmability
- Build Applications with Cisco: https://developer.cisco.com/learning/tracks/app-dev
- IOS-XE Programmability: https://developer.cisco.com/learning/tracks/iosxe-programmability
- Network Programmability for Network Engineers: https://developer.cisco.com/learning/tracks/netprog-eng

Key Topic

Getting Started with APIs

APIs are used everywhere these days. A large number of modern applications use some type of APIs because they make access available to other systems to interact with the application. There are few methods or technologies behind modern APIs:

- **Simple Object Access Protocol (SOAP):** SOAP is a standards-based web services access protocol that was originally developed by Microsoft and has been used by numerous legacy applications for many years. SOAP exclusively uses XML to provide API services. XML-based specifications are governed by XML Schema Definition (XSD) documents. SOAP was originally created to replace older solutions such as the Distributed Component Object Model (DCOM) and Common Object Request Broker Architecture (CORBA). You can find the latest SOAP specifications at https://www.w3.org/TR/soap.

- **Representational State Transfer (REST):** REST is an API standard that is easier to use than SOAP. It uses JSON instead of XML, and it uses standards like Swagger and the OpenAPI Specification (https://www.openapis.org) for ease of documentation and to help with adoption.
- **GraphQL and queryable APIs:** This is another query language for APIs that provides many developer tools. GraphQL is now used for many mobile applications and online dashboards. Many languages support GraphQL. You can learn more about GraphQL at https://graphql.org/code.

NOTE SOAP and REST share similarities over the HTTP protocol. SOAP limits itself to a stricter set of API messaging patterns than REST.

APIs often provide a roadmap describing the underlying implementation of an application. API documentation can provide a great level of detail that can be very valuable to security professional. These types of documentation include the following:

- **Swagger (OpenAPI):** Swagger is a modern framework of API documentation and is now the basis of the OpenAPI Specification (OAS). Additional information about Swagger can be obtained at https://swagger.io. The OAS specification is available at https://github.com/OAI/OpenAPI-Specification.
- **Web Services Description Language (WSDL) documents:** WSDL is an XML-based language that is used to document the functionality of a web service. The WSDL specification can be accessed at https://www.w3.org/TR/wsdl20-primer.
- **Web Application Description Language (WADL) documents:** WADL is also an XML-based language for describing web applications. The WADL specification can be obtained from https://www.w3.org/Submission/wadl.

NOTE Most Cisco products and services use RESTful (REST) APIs.

Key Topic

REST APIs

Let's take a look at a quick example of a REST API. There is a sample API you can use to perform several tests at https://deckofcardsapi.com. In Figure 3-29, the Linux **curl** utility is used to retrieve a "new deck of cards" from the Deck of Cards API. The API "shuffles" a deck of cards for you. The deck ID (**deck_id**) is **wkc12q20frlh** in this example.

NOTE The **python -m json.tool** command is used to invoke the json.tool Python module to "pretty print" the JSON output. You can obtain more information about the json.tool Python module at https://docs.python.org/3/library/json.html#module-json.tool.

Suppose that you want to draw a random card from the deck. Since you have the deck ID, you can easily use the command shown in Figure 3-30 to draw a random card.

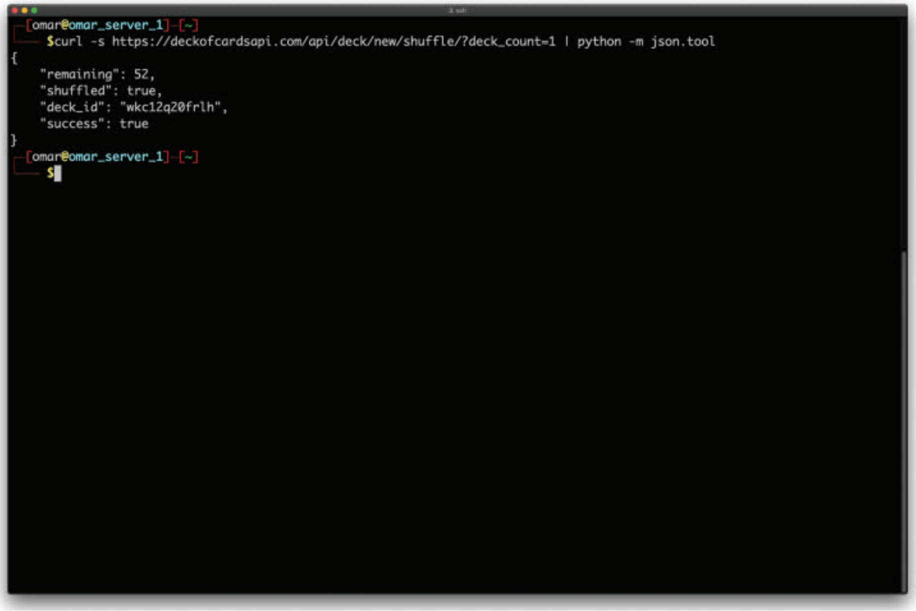


Figure 3-29 *Using **curl** to Obtain Information from an API*

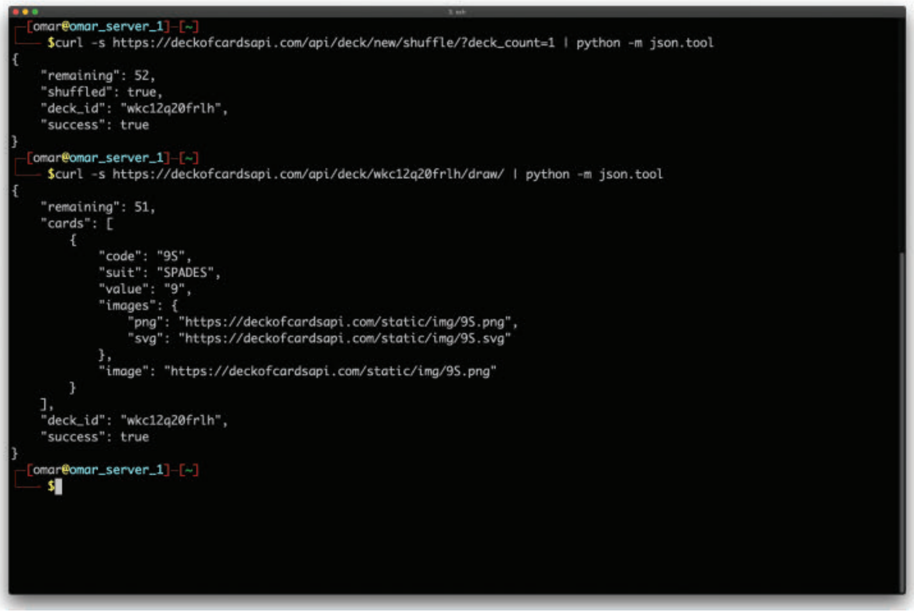


Figure 3-30 *Using **curl** to Obtain Additional Information from the Deck of Cards API*

You can see the response (in JSON), including the remaining number of cards and the card that was retrieved (the 9 of spades). Other information, such as the code, suit, value, and images of the card, is also included in the JSON output.

NOTE The DevNet tutorial at the following link shows how to interact with this sample API using Postman: https://developer.cisco.com/learning/lab/hands-on-postman/step/1.

Using Network Device APIs

3

Earlier in this chapter you learned that there are several API resources available in many Cisco solutions such as the Cisco DNA Center. The following are a few basic available API resources on the Cisco DNA Center Platform (10.1.1.1 is the IP address of the Cisco DNA Center):

- **https://10.1.1.1/api/system/v1/auth/token**: Used to get and encapsulate user identity and role information as a single value.
- **https://10.1.1.1/api/v1/network-device**: Used to get the list of first 500 network devices sorted lexicographically based on host name.
- **https://10.1.1.1/api/v1/interface**: Used to get information about every interface on every network device.
- **https://10.1.1.1/api/v1/host**: Used to get the name of a host, the ID of the VLAN that the host uses, the IP address of the host, the MAC address of the host, the IP address of the network device to which the host is connected, and more.
- **https://10.1.1.1/api/v1/flow-analysis**: Used to trace a path between two IP addresses. The function will wait for analysis to complete, and return the results.

There are a dozen (or dozens?) more APIs that you can use and interact with Cisco DNA Center at https://developer.cisco.com/dnacenter. Many other Cisco products include APIs that can be used for integrating third-party applications, obtain information similar to the preceding examples, as well as change the configuration of the device, apply policies, and more. Many of those APIs are also documented in DevNet (developer.cisco.com).

Modern networking devices support programmable capabilities such as NETCONF, RESTCONF, and YANG models. The following sections provide details about these technologies.

Key Topic

YANG Models

YANG is an API contract language used in many networking devices. In other words, you can use YANG to write a specification for what the interface between a client and networking device (server) should be on a particular topic. YANG was originally defined in RFC 6020 (https://tools.ietf.org/html/rfc6020).

TIP A specification written in YANG is referred to as a "YANG module." A collection (or set) of YANG modules are often called a "YANG model."

A YANG model typically concentrates on the data that a client processes using standardized operations.

NOTE Keep in mind that in NETCONF and RESTCONF implementations, the YANG controller is the client and the network elements are the server. You will learn more about NETCONF and RESTCONF later in this chapter.

Figure 3-31 shows an example of a network management application (client) interacting with a router (server) using YANG as the API contract.

Figure 3-31 *A Basic YANG Example*

A YANG-based server (as shown in Figure 3-31) publishes a set of YANG modules, which taken together form the system's YANG model. The YANG modules specify what a client can do. The following are a few examples of what a client can do using different YANG models:

- **Configure:** For example, enabling a routing protocol or a particular interface.
- **Receive notifications:** An example of notifications can be repeated login failures, interface failures, and so on.
- **Monitor status:** For example, retrieving information about CPU and memory utilization, packet counters, and so on.
- **Invoke actions:** For instance, resetting packet counters, rebooting the system, and so on.

NOTE The YANG model of a device is often called a "schema" defining the structure and content of messages exchanged between the application and the device.

The YANG language provides flexibility and extensibility capabilities that are not present in other model languages. When you create new YANG modules, you can leverage the data hierarchies defined in other modules. YANG also permits new statements to be defined, allowing the language itself to be expanded in a consistent way.

TIP DevNet has a series of videos that demonstrate how YANG works at https://developer.cisco.com/video/net-prog-basics/02-network_device_apis/yang.

Key Topic

NETCONF

NETCONF is defined in RFCs 6241 and 6242. NETCONF was created to overcome the challenges in legacy Simple Network Management Protocol (SNMP) implementations.

A NETCONF client typically has the role of a network management application. The NETCONF server is a managed network device (router, switch, and so on). You can also have intermediate systems (often called “controllers”) that control a particular aspect or domain. Controllers can act as a server to its managers and as a client to its networking devices, as shown in Figure 3-32.

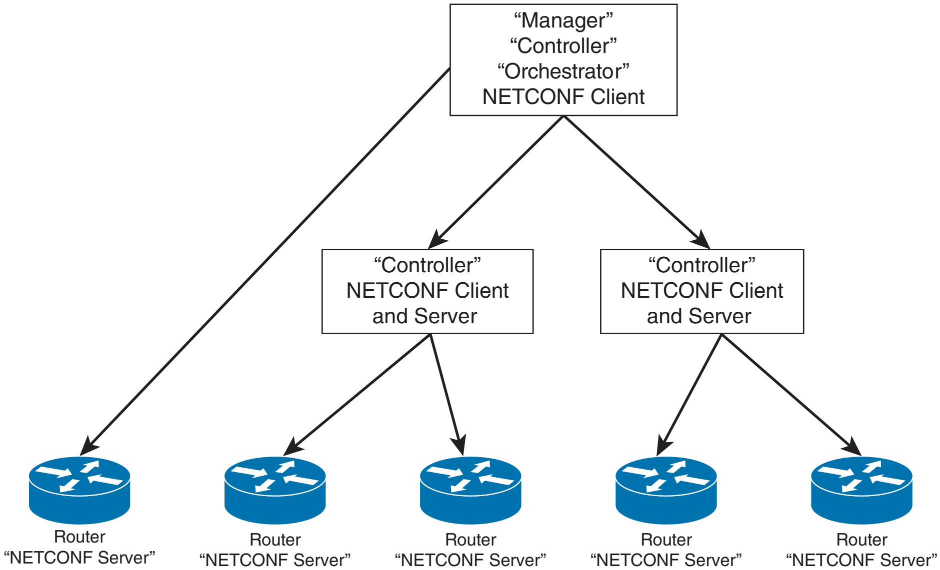


Figure 3-32 *NETCONF Clients, Servers, and Controllers*

In Figure 3-32, a node called a “Manager” manages a NETCONF server (router) and two “Controllers,” which are both a server for the Manager and a client for the other network devices (routers).

NOTE NETCONF was created before YANG. Other languages were used for NETCONF operations. On the other hand, YANG is the only language widely used for NETCONF nowadays.

NETCONF sessions established from a NETCONF client to a NETCONF server consist of a sequence of messages. Both parties send a “hello” message when they initially connect. All message exchanges are initiated by the NETCONF client. The hello message includes which NETCONF protocol version(s) the devices support. The server states which optional capabilities it supports.

NETCONF messages are either a remote procedure call (RPC) or an “rpc-reply.” Each RPC is a request from the client to the server to execute a given operation. The NETCONF rpc-reply is sent by the server when it has completed or failed to complete the request. Some NETCONF rpc-replies are short answers to a simple query, or just an OK that the order was executed. Some are long and may contain the entire device configuration or status. NETCONF rpc-replies to subscriptions consist of a message that technically never ends. Other information of the rpc-reply is generated by the server. A NETCONF rpc-reply may also be a NETCONF rpc-error, indicating that the requested operation failed.

NETCONF messages are encoded in an XML-based structure defined by the NETCONF standard. The NETCONF communication is done over Secure Shell (SSH), but using a default TCP port 830. This can be configured to a different port.

SSH supports a subsystem concept. NETCONF has its own subsystem: netconf. Figure 3-33 shows how you can connect to a networking device (in this case, a CSR-1000v router configured with the hostname **ios-xe-mgmt.cisco.com**). The username of the router is **root**. You are also asked to provide a password. The router is configured for NETCONF over TCP port 10000.

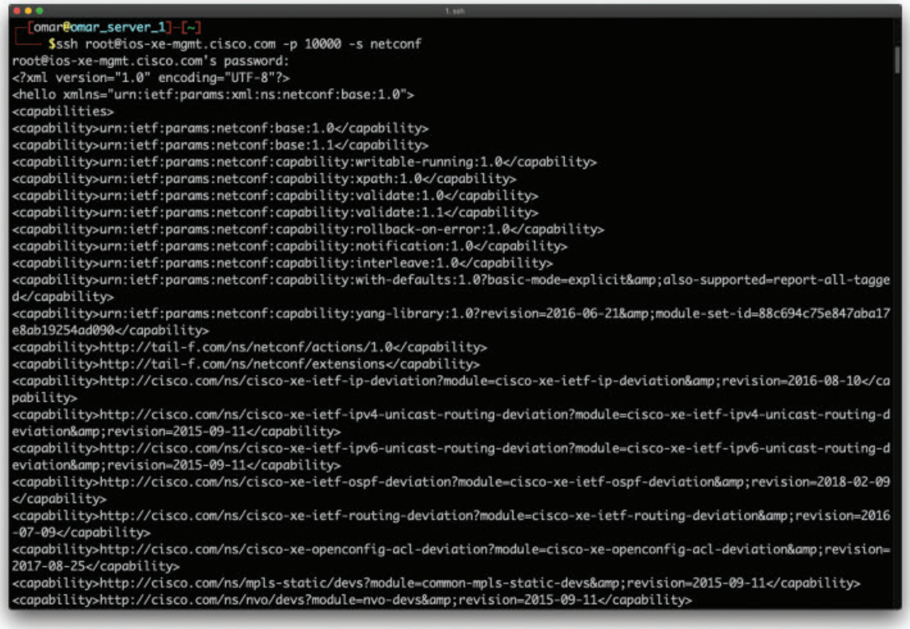


Figure 3-33 *Using the NETCONF SSH Subsystem*

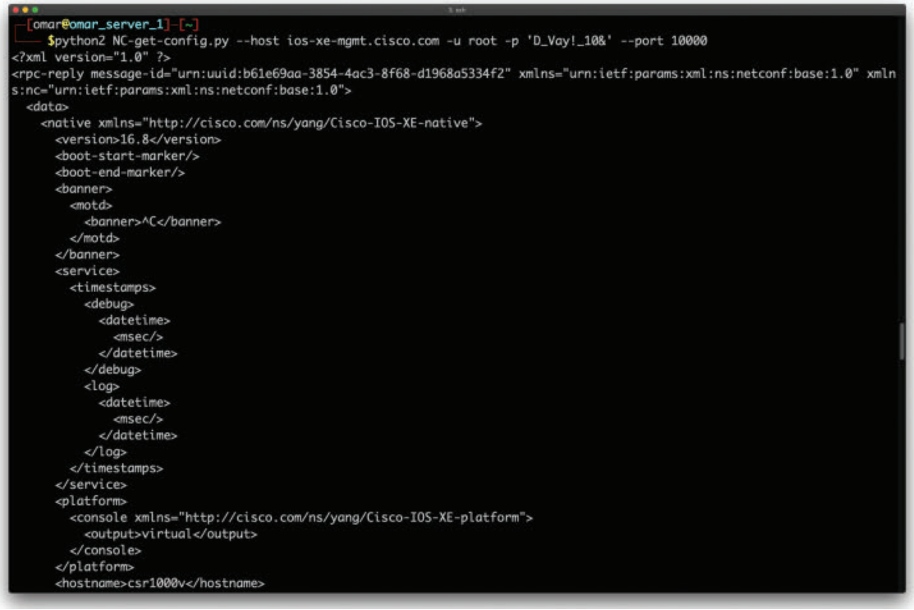
TIP DevNet has several sandboxes where you can practice these concepts and more at https://devnetsandbox.cisco.com.

An open source Python library for NETCONF clients called ncclient is available on GitHub at https://github.com/ncclient/ncclient. You can install it using Python pip, as shown here:

```
pip install ncclient
```

There are several sample scripts at the DevNet GitHub repositories that can help you get started at https://github.com/CiscoDevNet/python_code_samples_network.

Figure 3-34 shows how to use a Python script that leverages ncclient to interact with the router (**ios-xe-mgmt.cisco.com**).



```
┌─[omar@omar_server_1]─[~]
└──── $python2 NC-get-config.py --host ios-xe-mgmt.cisco.com -u root -p 'D_Vay!_10&' --port 10000
<?xml version="1.0" ?>
<rpc-reply message-id="urn:uuid:b61e69aa-3854-4ac3-8f68-d1968a5334f2" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmln
s:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
      <version>16.8</version>
      <boot-start-marker/>
      <boot-end-marker/>
      <banner>
        <motd>
          <banner>^C</banner>
        </motd>
      </banner>
      <service>
        <timestamps>
          <debug>
            <datetime>
              <msec/>
            </datetime>
          </debug>
          <log>
            <datetime>
              <msec/>
            </datetime>
          </log>
        </timestamps>
      </service>
      <platform>
        <console xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-platform">
          <output>virtual</output>
        </console>
      </platform>
      <hostname>csr1000v</hostname>
```

Figure 3-34 *Using Python to Obtain the Entire Configuration of a Network Device*

TIP You can obtain NC-get-config.py from https://github.com/CiscoDevNet/python_code_samples_network/tree/master/NC-get-config.

Key Topic

RESTCONF

You already learned that REST is a type of modern API. Many network administrators wanted to have the capabilities of NETCONF over "REST." This is why a REST-based variant of NETCONF was created. RESTCONF is now supported in many networking devices in the industry.

RESTCONF is defined in RFC 8040 and it follows the REST principles. However, not all REST-based APIs are compatible or even comparable to RESTCONF.

The RESTCONF interface is built around a small number of standardized requests (GET, PUT, POST, PATCH, and DELETE). Several of the REST principles are similar to NETCONF:

- The client-server model
- The layered system principle
- The first two uniform interface principles

One of the differences between RESTCONF and NETCONF is the stateless server principle. NETCONF is based on clients establishing a session to the server (which is not stateless). NETCONF clients frequently connect and then manipulate the candidate datastore with a number of *edit-config* operations. The NETCONF clients may also send a *validation* call to NETCONF servers. This is different in RESTCONF.

RESTCONF requires the server to keep some client state. Any request the RESTCONF client sends is acted upon by the server immediately. You cannot send any transactions that span multiple RESTCONF messages. Subsequently, some of the key features of NETCONF (including networkwide transactions) are not possible in RESTCONF.

Let's take a look at a quick example of using RESTCONF. Example 3-1 shows a Python script that is used to obtain the details of all interfaces in a networking device using RESTCONF.

Example 3-1 *Python Script to Retrieve Interface Details from a Networking Device Using RESTCONF*

```
#!/usr/bin/python
import requests
import sys

# disable warnings from SSL/TLS certificates
requests.packages.urllib3.disable_warnings()

# the IP address or hostname of the networking device
HOST = 'ios-xe-mgmt.cisco.com'

# use your user credentials to access the networking device
USER = 'root'
PASS = ‘supersecretpassword'

# create a main() method
def main():
    """Main method that retrieves the interface details from a
    networking device via RESTCONF."""

    # RESTCONF url of the networking device
    url="https://{h}:9443/restconf/data/ietf-
    interfaces:interfaces".format(h=HOST)

    # RESTCONF media types for REST API headers
    headers = {'Content-Type': 'application/yang-data+json',
               'Accept': 'application/yang-data+json'}

    # this statement performs a GET on the specified url
    response = requests.get(url, auth=(USER, PASS),
                            headers=headers, verify=False)

    # print the json that is returned
    print(response.text)

if __name__ == '__main__':
    sys.exit(main())
```

Figure 3-35 shows the output of the Python script, including the information of all the interfaces in that networking device (**ios-xe-mgmt.cisco.com**).

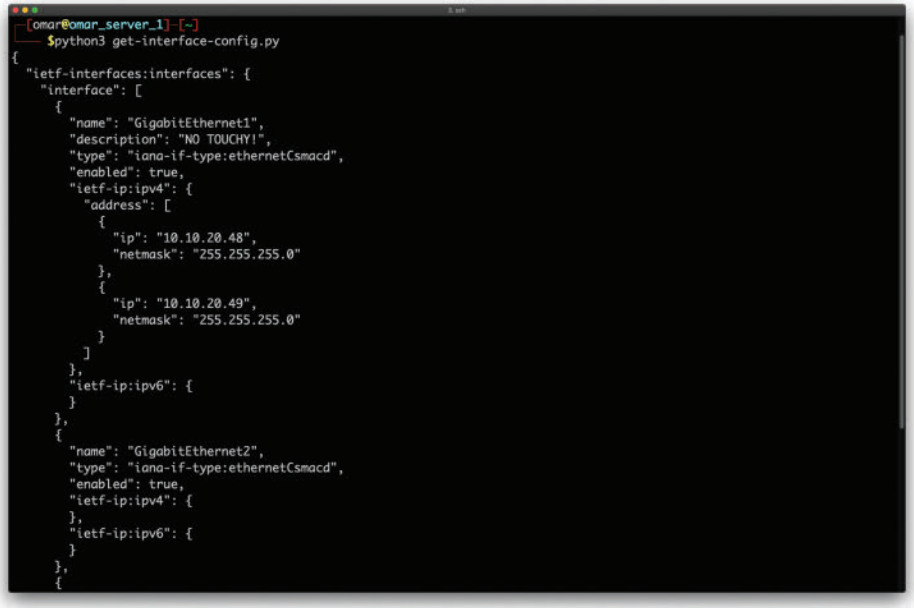


Figure 3-35 *Using Python to Obtain Information from a Network Device Using RESTCONF*

TIP Watch the DevNet "Getting Started with Network Device APIs" video for additional step-by-step information about Network APIs, NETCONF, RESTCONF, and YANG at https://developer.cisco.com/video/net-prog-basics/02-network_device_apis.

OpenConfig and gNMI

The OpenConfig consortium (https://github.com/openconfig) is a collaborative effort to provide vendor-neutral data models (in YANG) for network devices. OpenConfig uses the gRPC Network Management Interface (gNMI). The following GitHub repository includes detailed information about gNMI, as well as sample code (https://github.com/openconfig/gnmi).

NOTE The gRPC specification (https://grpc.io) is a modern Remote Procedure Call (RPC) framework. RPC allows a client to invoke operations (also called "procedures") on a server. RPC includes an interface description language (IDL) used to state what procedures the server supports (including the input and output data from them). RPC also uses client libraries to call upon those procedures (supported in different programming languages). RPC uses a serialization, marshalling, and transport mechanism for the messages (generally called an RPC protocol).

The gNMI protocol is similar to NETCONF and RESTCONF. gNMI uses YANG models, but it can be used with other interface description languages (IDLs). The OpenConfig consortium defined several standard YANG models to go with the protocols. These YANG models describe many essential networking features such as interface configuration, routing protocols, QoS, Wi-Fi configurations, and more.

Exam Preparation Tasks

As mentioned in the section "How to Use This Book" in the Introduction, you have a couple of choices for exam preparation: the exercises here, Chapter 12, "Final Preparation," and the exam simulation questions in the Pearson Test Prep Software Online.

Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. Table 3-2 lists these key topics and the page numbers on which each is found.

Key Topic

Table 3-2 Key Topics for Chapter 3

Key Topic Element	Description	Page Number
Section	Traditional Networking Planes	109
Section	So What's Different with SDN?	110
Section	Introduction to the Cisco ACI Solution	110
List	Understand the functions of the APIC	112
Section	VXLAN and Network Overlays	112
Paragraph	Understand what is micro-segmentation	115
Paragraph	Understand "east-west" traffic and "north-south" traffic	115
Section	Open Source Initiatives	117
Paragraph	Understand northbound and southbound APIs	118
Section	More About Network Function Virtualization	118
Section	Cisco DNA Center APIs	130
Tip	Cisco DNA Center APIs in DevNet	130
Section	Cisco DNA Security Solution	132
Section	Modern Programming Languages and Tools	133
Section	DevNet	136
Section	Getting Started with APIs	136
Section	REST APIs	137
Section	YANG Models	139
Section	NETCONF	141
Section	RESTCONF	143

Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

Representational State Transfer (REST), Simple Object Access Protocol (SOAP), Contiv, Network Functions Virtualization (NFV), Neutron, Open vSwitch, OpenDaylight (ODL), YANG, NETCONF, RESTCONF

Review Questions

1. The RESTCONF interface is built around a small number of standardized requests. Which of the following are requests supported by RESTCONF?
   a. GET
   b. PUT
   c. PATCH
   d. All of these answers are correct.
2. NETCONF messages are encoded in a(n) ________ structure defined by the NETCONF standard.
   a. JSON
   b. XML
   c. OWASP
   d. RESTCONF
3. Which of the following is a Cisco resource where you can learn about network programmability and obtain sample code?
   a. APIC
   b. ACI
   c. DevNet
   d. NETCONF
4. A YANG-based server publishes a set of YANG modules, which taken together form the system's _________.
   a. YANG model
   b. NETCONF model
   c. RESTCONF model
   d. gRPC model
5. Which of the following HTTP methods sends data to the server typically used in HTML forms and API requests?
   a. POST
   b. GET
   c. TRACE
   d. PUT

6. Which of the following is a solution that allows you to detect security threats in encrypted traffic without decrypting the packets?
   a. ETA
   b. ESA
   c. WSA
   d. None of these answers is correct.
7. Which of the following is an open source project that allows you to deploy micro-segmentation policy-based services in container environments?
   a. OVS
   b. Contiv
   c. ODL
   d. All of the above
8. NFV nodes such as virtual routers and firewalls need which of the following components as an underlying infrastructure?
   a. A hypervisor
   b. A virtual forwarder to connect individual instances
   c. A network controller
   d. All of these answers are correct.
9. There have been multiple IP tunneling mechanisms introduced throughout the years. Which of the following are examples of IP tunneling mechanisms?
   a. VXLAN
   b. SST
   c. NVGRE
   d. All of these answers are correct.
10. Which of the following is true about SDN?
    a. SDN provides numerous benefits in the area of management plane. These benefits are in both physical switches and virtual switches.
    b. SDN changed a few things in the management, control, and data planes. However, the big change was in the control and data planes in software-based switches and routers (including virtual switches inside of hypervisors).
    c. SDN is now widely adopted in data centers.
    d. All of these answers are correct.

Index

Numbers

A

B

C

D

F

G

H

I

J - K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y - Z