



Practice Tests



Video Training



Flash Cards

Official Cert Guide

Advance your IT career with hands-on learning

CCNP and CCIE Security Core SCOR 350-701



Study Planner



Review Exercises

ciscopress.com

OMAR SANTOS

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

Copyright © 2020 Cisco Systems, Inc.

Published by:

Cisco Press

221 River St.

Hoboken, NJ 07030 USA

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions.

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

ScoutAutomatedPrintCode

Library of Congress Control Number: 2020901233

ISBN-10: 0-13-597197-7

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

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

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
- 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

	Introduction	xxv
Chapter 1	Cybersecurity Fundamentals	2
Chapter 2	Cryptography	78
Chapter 3	Software-Defined Networking Security and Network Programmability	106
Chapter 4	Authentication, Authorization, Accounting (AAA) and Identity Management	150
Chapter 5	Network Visibility and Segmentation	220
Chapter 6	Infrastructure Security	306
Chapter 7	Cisco Next-Generation Firewalls and Cisco Next-Generation Intrusion Prevention Systems	392
Chapter 8	Virtual Private Networks (VPNs)	464
Chapter 9	Securing the Cloud	548
Chapter 10	Content Security	600
Chapter 11	Endpoint Protection and Detection	634
Chapter 12	Final Preparation	658
	Glossary of Key Terms	660
Appendix A	Answers to the “Do I Know This Already?” Quizzes and Q&A Sections	678
Appendix B	CCNP Security Core SCOR (350-701) Exam Updates	686
	Index	688

Contents

	Introduction	xxv
Chapter 1	Cybersecurity Fundamentals	2
	“Do I Know This Already?” Quiz	3
	Foundation Topics	6
	Introduction to Cybersecurity	6
	Cybersecurity vs. Information Security (InfoSec)	7
	The NIST Cybersecurity Framework	7
	Additional NIST Guidance and Documents	7
	The International Organization for Standardization (ISO)	8
	Defining What Are Threats, Vulnerabilities, and Exploits	8
	What Is a Threat?	9
	What Is a Vulnerability?	9
	What Is an Exploit?	10
	Risk, Assets, Threats, and Vulnerabilities	12
	Defining Threat Actors	13
	Understanding What Threat Intelligence Is	14
	Viruses and Worms	16
	<i>Types and Transmission Methods</i>	16
	<i>Malware Payloads</i>	17
	Trojans	18
	<i>Trojan Types</i>	18
	<i>Trojan Ports and Communication Methods</i>	19
	<i>Trojan Goals</i>	20
	<i>Trojan Infection Mechanisms</i>	20
	<i>Effects of Trojans</i>	22
	Distributing Malware	22
	Ransomware	23
	Covert Communication	23
	Keyloggers	25
	Spyware	26
	Analyzing Malware	27
	Static Analysis	27
	Dynamic Analysis	28

Common Software and Hardware Vulnerabilities	30
Injection Vulnerabilities	30
SQL Injection	30
HTML Injection	32
Command Injection	32
Authentication-based Vulnerabilities	32
<i>Credential Brute Force Attacks and Password Cracking</i>	33
<i>Session Hijacking</i>	34
<i>Default Credentials</i>	34
<i>Insecure Direct Object Reference Vulnerabilities</i>	35
Cross-site Scripting (XSS)	35
Cross-site Request Forgery	37
Cookie Manipulation Attacks	37
Race Conditions	38
Unprotected APIs	38
Return-to-LibC Attacks and Buffer Overflows	39
OWASP Top 10	40
Security Vulnerabilities in Open Source Software	40
Confidentiality, Integrity, and Availability	40
What Is Confidentiality?	40
What Is Integrity?	42
What Is Availability?	43
Talking About Availability, What Is a Denial-of-Service (DoS) Attack?	44
Access Control Management	45
Cloud Security Threats	47
Cloud Computing Issues and Concerns	48
Cloud Computing Attacks	50
Cloud Computing Security	51
IoT Security Threats	51
IoT Protocols	53
Hacking IoT Implementations	54
An Introduction to Digital Forensics and Incident Response	55
ISO/IEC 27002:2013 and NIST Incident Response Guidance	55
What Is an Incident?	56
False Positives, False Negatives, True Positives, and True Negatives	57
Incident Severity Levels	58
How Are Incidents Reported?	58

What Is an Incident Response Program?	60
The Incident Response Plan	60
The Incident Response Process	61
Tabletop Exercises and Playbooks	63
Information Sharing and Coordination	64
Computer Security Incident Response Teams	64
Product Security Incident Response Teams (PSIRTs)	66
The Common Vulnerability Scoring System (CVSS)	67
National CSIRTs and Computer Emergency Response Teams (CERTs)	71
Coordination Centers	72
Incident Response Providers and Managed Security Service Providers (MSSPs)	73
Key Incident Management Personnel	73
Summary	74
Exam Preparation Tasks	74
Review All Key Topics	74
Define Key Terms	76
Review Questions	76
Chapter 2 Cryptography	78
“Do I Know This Already?” Quiz	78
Foundation Topics	80
Introduction to Cryptography	80
Ciphers	80
Keys	81
Block and Stream Ciphers	82
Symmetric and Asymmetric Algorithms	82
Hashes	84
Hashed Message Authentication Code	86
Digital Signatures	86
Key Management	89
Next-Generation Encryption Protocols	89
IPsec	90
SSL and TLS	91
Fundamentals of PKI	93
Public and Private Key Pairs	93
More About Keys and Digital Certificates	93
Certificate Authorities	94
Root Certificates	95

	Identity Certificates	96
	X.500 and X.509v3	97
	Authenticating and Enrolling with the CA	98
	Public Key Cryptography Standards	99
	Simple Certificate Enrollment Protocol	99
	Revoking Digital Certificates	99
	Digital Certificates in Practice	100
	PKI Topologies	101
	<i>Single Root CA</i>	101
	<i>Hierarchical CA with Subordinate CAs</i>	101
	<i>Cross-Certifying CAs</i>	102
	Exam Preparation Tasks	102
	Review All Key Topics	102
	Define Key Terms	103
	Review Questions	103
Chapter 3	Software-Defined Networking Security and Network Programmability	106
	“Do I Know This Already?” Quiz	106
	Foundation Topics	108
	Introduction to Software-Defined Networking	108
	Traditional Networking Planes	109
	So What’s Different with SDN?	110
	Introduction to the Cisco ACI Solution	110
	VXLAN and Network Overlays	112
	Micro-Segmentation	115
	Open Source Initiatives	117
	More About Network Function Virtualization	118
	NFV MANO	119
	Contiv	120
	Cisco Digital Network Architecture (DNA)	121
	Cisco DNA Policies	123
	Cisco DNA Group-Based Access Control Policy	124
	Cisco DNA IP-Based Access Control Policy	126
	Cisco DNA Application Policies	126
	Cisco DNA Traffic Copy Policy	127
	Cisco DNA Center Assurance Solution	128
	Cisco DNA Center APIs	130
	Cisco DNA Security Solution	132

Cisco DNA Multivendor Support	132
Introduction to Network Programmability	132
Modern Programming Languages and Tools	133
DevNet	136
Getting Started with APIs	136
REST APIs	137
Using Network Device APIs	139
YANG Models	139
NETCONF	141
RESTCONF	143
OpenConfig and gNMI	145
Exam Preparation Tasks	146
Review All Key Topics	146
Define Key Terms	147
Review Questions	147

Chapter 4 Authentication, Authorization, Accounting (AAA) and Identity Management 150

“Do I Know This Already?” Quiz	151
Foundation Topics	154
Introduction to Authentication, Authorization, and Accounting	154
The Principle of Least Privilege and Separation of Duties	155
Authentication	155
Authentication by Knowledge	156
Authentication by Ownership or Possession	157
Authentication by Characteristic	158
Multifactor Authentication	159
Duo Security	159
Zero Trust and BeyondCorp	161
Single Sign-On	164
Authorization	167
Mandatory Access Control (MAC)	168
Discretionary Access Control (DAC)	168
Role-Based Access Control (RBAC)	168
Rule-Based Access Control	169
Attribute-Based Access Control	169
Accounting	169
Infrastructure Access Controls	170
Access Control Mechanisms	170

AAA Protocols	172
RADIUS	173
TACACS+	174
Diameter	176
802.1X	178
Network Access Control List and Firewalling	180
VLAN ACLs	181
Security Group–Based ACL	181
Downloadable ACL	181
Cisco Identity Services Engine (ISE)	181
Cisco Platform Exchange Grid (pxGrid)	182
Cisco ISE Context and Identity Services	184
Cisco ISE Profiling Services	184
Cisco ISE Identity Services	187
Cisco ISE Authorization Rules	188
Cisco TrustSec	190
Posture Assessment	192
Change of Authorization (CoA)	193
Configuring TACACS+ Access	196
Configuring RADIUS Authentication	202
Configuring 802.1X Authentication	205
Additional Cisco ISE Design Tips	211
Advice on Sizing a Cisco ISE Distributed Deployment	214
Exam Preparation Tasks	214
Review All Key Topics	214
Define Key Terms	216
Review Questions	216
Chapter 5 Network Visibility and Segmentation	220
“Do I Know This Already?” Quiz	221
Foundation Topics	224
Introduction to Network Visibility	224
NetFlow	225
The Network as a Sensor and as an Enforcer	226
What Is a Flow?	227
NetFlow for Network Security and Visibility	229
NetFlow for Anomaly Detection and DDoS Attack Mitigation	229
Data Leak Detection and Prevention	231

Incident Response, Threat Hunting, and Network Security Forensics	231
Traffic Engineering and Network Planning	236
NetFlow Versions	237
IP Flow Information Export (IPFIX)	237
IPFIX Architecture	238
Understanding IPFIX Mediators	239
IPFIX Templates	239
Option Templates	241
Understanding the Stream Control Transmission Protocol (SCTP)	241
Exploring Application Visibility and Control and NetFlow	241
Application Recognition	241
Metrics Collection and Exporting	242
NetFlow Deployment Scenarios	242
NetFlow Deployment Scenario: User Access Layer	243
NetFlow Deployment Scenario: Wireless LAN	244
NetFlow Deployment Scenario: Internet Edge	245
NetFlow Deployment Scenario: Data Center	246
NetFlow Deployment Scenario: NetFlow in Site-to-Site and Remote VPNs	248
Cisco Stealthwatch	250
Stealthwatch Cloud	251
On-Premises Monitoring with Cisco Stealthwatch Cloud	256
Cisco Stealthwatch Cloud Integration with Meraki and Cisco Umbrella	256
Exploring the Cisco Stealthwatch On-Premises Appliances	256
Threat Hunting with Cisco Stealthwatch	258
Cisco Cognitive Threat Analytics (CTA) and Encrypted Traffic Analytics (ETA)	262
What Is Cisco ETA?	262
What Is Cisco Cognitive Threat Analytics?	262
NetFlow Collection Considerations and Best Practices	268
Determining the Flows per Second and Scalability	269
Configuring NetFlow in Cisco IOS and Cisco IOS-XE	269
Simultaneous Application Tracking	270
Flexible NetFlow Records	271
Flexible NetFlow Key Fields	271
Flexible NetFlow Non-Key Fields	273
NetFlow Predefined Records	274

User-Defined Records	275
Flow Monitors	275
Flow Exporters	275
Flow Samplers	275
Flexible NetFlow Configuration	275
Configure a Flow Record	276
Configure a Flow Monitor for IPv4 or IPv6	278
Configure a Flow Exporter for the Flow Monitor	280
Apply a Flow Monitor to an Interface	282
Flexible NetFlow IPFIX Export Format	283
Configuring NetFlow in NX-OS	283
Introduction to Network Segmentation	285
Data-Driven Segmentation	286
Application-Based Segmentation	288
Micro-Segmentation with Cisco ACI	289
Segmentation with Cisco ISE	290
The Scalable Group Tag Exchange Protocol (SXP)	292
SGT Assignment and Deployment	294
Initially Deploying 802.1X and/or TrustSec in Monitor Mode	294
Active Policy Enforcement	295
Cisco ISE TrustSec and Cisco ACI Integration	298
Exam Preparation Tasks	301
Review All Key Topics	301
Define Key Terms	302
Review Questions	302
Chapter 6 Infrastructure Security	306
“Do I Know This Already?” Quiz	307
Foundation Topics	310
Securing Layer 2 Technologies	310
VLAN and Trunking Fundamentals	310
What Is a VLAN?	311
Trunking with 802.1Q	313
Let’s Follow the Frame, Step by Step	315
What Is the Native VLAN on a Trunk?	315
So, What Do You Want to Be? (Asks the Port)	316
Understanding Inter-VLAN Routing	316
What Is the Challenge of Only Using Physical Interfaces?	316

Using Virtual “Sub” Interfaces	316
Spanning Tree Fundamentals	317
The Solution to the Layer 2 Loop	318
STP Is Wary of New Ports	321
Improving the Time Until Forwarding	321
Common Layer 2 Threats and How to Mitigate Them	322
Do Not Allow Negotiations	323
Layer 2 Security Toolkit	324
BPDU Guard	324
Root Guard	325
Port Security	325
CDP and LLDP	327
DHCP Snooping	328
Dynamic ARP Inspection	330
Network Foundation Protection	332
The Importance of the Network Infrastructure	332
The Network Foundation Protection Framework	333
Interdependence	333
Implementing NFP	333
Understanding and Securing the Management Plane	334
Best Practices for Securing the Management Plane	334
Understanding the Control Plane	336
Best Practices for Securing the Control Plane	336
Understanding and Securing the Data Plane	337
Best Practices for Protecting the Data Plane	337
Additional Data Plane Protection Mechanisms	338
Securing Management Traffic	338
What Is Management Traffic and the Management Plane?	338
Beyond the Console Cable	339
Management Plane Best Practices	339
Password Recommendations	341
Using AAA to Verify Users	342
Router Access Authentication	342
The AAA Method List	343
Role-Based Access Control	344
Custom Privilege Levels	344
Limiting the Administrator by Assigning a View	344

Encrypted Management Protocols	344
Using Logging Files	345
Understanding NTP	346
Protecting Cisco IOS, Cisco IOS-XE, Cisco IOS-XR, and Cisco NX-OS Files	346
Implementing Security Measures to Protect the Management Plane	347
Implementing Strong Passwords	347
User Authentication with AAA	349
Using the CLI to Troubleshoot AAA for Cisco Routers	353
RBAC Privilege Level/Parser View	356
Implementing Parser Views	358
SSH and HTTPS	360
Implementing Logging Features	362
Configuring Syslog Support	363
Configuring NTP	363
Securing the Network Infrastructure Device Image and Configuration Files	364
Securing the Data Plane in IPv6	365
Understanding and Configuring IPv6	365
The Format of an IPv6 Address	367
Understanding the Shortcuts	367
Did We Get an Extra Address?	367
IPv6 Address Types	368
Configuring IPv6 Routing	370
Moving to IPv6	372
Developing a Security Plan for IPv6	372
Best Practices Common to Both IPv4 and IPv6	372
Threats Common to Both IPv4 and IPv6	373
The Focus on IPv6 Security	374
New Potential Risks with IPv6	375
IPv6 Best Practices	376
IPv6 Access Control Lists	377
Securing Routing Protocols and the Control Plane	379
Minimizing the Impact of Control Plane Traffic on the CPU	379
Details about CoPP	380
Details about CPPr	383
Securing Routing Protocols	383

	Implementing Routing Update Authentication on OSPF	383
	Implementing Routing Update Authentication on EIGRP	384
	Implementing Routing Update Authentication on RIP	385
	Implementing Routing Update Authentication on BGP	386
	Exam Preparation Tasks	387
	Review All Key Topics	387
	Define Key Terms	389
	Review Questions	389
Chapter 7	Cisco Next-Generation Firewalls and Cisco Next-Generation Intrusion Prevention Systems	392
	“Do I Know This Already?” Quiz	392
	Foundation Topics	395
	Introduction to Cisco Next-Generation Firewalls (NGFW) and Next-Generation Intrusion Prevention Systems (NGIPS)	395
	Cisco Firewall History and Legacy	396
	Introducing the Cisco ASA	396
	The Cisco ASA FirePOWER Module	397
	Cisco Firepower Threat Defense (FTD)	397
	Cisco Firepower 1000 Series	397
	Cisco Firepower 2100 Series	397
	Cisco Firepower 4100 Series	398
	Cisco Firepower 9300 Series	399
	Cisco FTD for Cisco Integrated Services Routers (ISRs)	399
	Introduction to Cisco’s NGIPS	399
	Surveying the Cisco Firepower Management Center (FMC)	401
	Exploring the Cisco Firepower Device Manager (FDM)	404
	Cisco Defense Orchestrator	408
	Comparing Network Security Solutions That Provide Firewall Capabilities	411
	Deployment Modes of Network Security Solutions and Architectures That Provide Firewall Capabilities	412
	Routed vs. Transparent Firewalls	413
	Security Contexts	414
	Single-Mode Transparent Firewalls	414
	Surveying the Cisco FTD Deployment Modes	416
	Cisco FTD Interface Modes	417
	Inline Pair	420
	Inline Pair with Tap	420

Passive Mode	420
Passive with ERSPAN Mode	422
Additional Cisco FTD Deployment Design Considerations	422
High Availability and Clustering	423
Clustering	425
Implementing Access Control	427
Implementing Access Control Lists in Cisco ASA	427
Cisco ASA Application Inspection	433
To-the-Box Traffic Filtering in the Cisco ASA	434
Object Grouping and Other ACL Features	435
Standard ACLs	436
Time-Based ACLs	436
ICMP Filtering in the Cisco ASA	437
Network Address Translation in Cisco ASA	437
Cisco ASA Auto NAT	443
Implementing Access Control Policies in the Cisco Firepower Threat Defense	443
Cisco Firepower Intrusion Policies	446
Variables	449
Platform Settings Policy	450
Cisco NGIPS Preprocessors	450
Cisco Advanced Malware Protection (AMP)	452
Security Intelligence, Security Updates, and Keeping Firepower Software Up to Date	457
Security Intelligence Updates	457
Keeping Software Up to Date	458
Exam Preparation Tasks	458
Review All Key Topics	458
Define Key Terms	460
Review Questions	460
Chapter 8 Virtual Private Networks (VPNs)	464
“Do I Know This Already?” Quiz	464
Foundation Topics	467
Virtual Private Network (VPN) Fundamentals	467
An Overview of IPsec	470
IKEv1 Phase 1	470
IKEv1 Phase 2	472
NAT Traversal (NAT-T)	474

IKEv2	475
SSL VPNs	476
Cisco AnyConnect Secure Mobility	478
Deploying and Configuring Site-to-Site VPNs in Cisco Routers	479
Traditional Site-to-Site VPNs in Cisco IOS and Cisco IOS-XE Devices	479
Tunnel Interfaces	482
GRE over IPsec	482
More About Tunnel Interfaces	484
Multipoint GRE (mGRE) Tunnels	486
DMVPN	486
GETVPN	489
FlexVPN	492
Debug and Show Commands to Verify and Troubleshoot IPsec Tunnels	496
Configuring Site-to-Site VPNs in Cisco ASA Firewalls	502
Step 1: Enable ISAKMP in the Cisco ASA	503
Step 2: Create the ISAKMP Policy	503
Step 3: Set Up the Tunnel Groups	504
Step 4: Define the IPsec Policy	505
Step 5: Create the Crypto Map in the Cisco ASA	506
Step 6: Configure Traffic Filtering (Optional)	508
Step 7: Bypass NAT (Optional)	508
Step 8: Enable Perfect Forward Secrecy (Optional)	509
Additional Attributes in Cisco Site-to-Site VPN Configurations	509
Configuring Remote Access VPNs in the Cisco ASA	511
Configuring IPsec Remote Access VPN in the Cisco ASA	512
Configuring Clientless Remote Access SSL VPNs in the Cisco ASA	514
Cisco ASA Remote-Access VPN Design Considerations	515
Pre-SSL VPN Configuration Steps	516
Understanding the Remote Access VPN Attributes and Policy Inheritance Model	518
Configuring Clientless SSL VPN Group Policies	518
Configuring the Tunnel Group for Clientless SSL VPN	519
Configuring User Authentication for Clientless SSL VPN	520
Enabling Clientless SSL VPN	522
Configuring WebType ACLs	523
Configuring Application Access in Clientless SSL VPNs	524

Configuring Client-Based Remote-Access SSL VPNs in the Cisco ASA	525
Setting Up Tunnel and Group Policies	525
Deploying the AnyConnect Client	527
Understanding Split Tunneling	528
Understanding DTLS	529
Configuring Remote Access VPNs in FTD	530
Using the Remote Access VPN Policy Wizard	531
Troubleshooting Cisco FTD Remote Access VPN Implementations	540
Configuring Site-to-Site VPNs in FTD	541
Exam Preparation Tasks	543
Review All Key Topics	543
Define Key Terms	544
Review Questions	544
Chapter 9 Securing the Cloud	548
“Do I Know This Already?” Quiz	549
Foundation Topics	551
What Is Cloud and What Are the Cloud Service Models?	551
DevOps, Continuous Integration (CI), Continuous Delivery (CD), and DevSecOps	552
The Waterfall Development Methodology	552
The Agile Methodology	553
DevOps	556
CI/CD Pipelines	558
The Serverless Buzzword	559
Container Orchestration	559
A Quick Introduction to Containers and Docker	561
Kubernetes	565
Microservices and Micro-Segmentation	570
DevSecOps	571
Describing the Customer vs. Provider Security Responsibility for the Different Cloud Service Models	573
Patch Management in the Cloud	575
Security Assessment in the Cloud and Questions to Ask Your Cloud Service Provider	575
Cisco Umbrella	577
The Cisco Umbrella Architecture	577
Secure Internet Gateway	578
Cisco Umbrella Investigate	580

Cisco Email Security in the Cloud	582
Forged Email Detection	583
Sender Policy Framework	583
Email Encryption	583
Cisco Email Security for Office 365	583
Cisco Cloudlock	584
Stealthwatch Cloud	590
AppDynamics Cloud Monitoring	590
Cisco Tetration	593
Tetration Agents	593
Application Dependency Mapping	594
Tetration Forensics Feature	594
Tetration Security Dashboard	594
Exam Preparation Tasks	596
Review All Key Topics	596
Define Key Terms	597
Review Questions	598

Chapter 10 Content Security 600

“Do I Know This Already?” Quiz	600
Foundation Topics	603
Content Security Fundamentals	603
Cisco Async Operating System (AsyncOS)	604
Cisco WSA	604
The Cisco WSA Proxy	605
Cisco WSA in Explicit Forward Mode	606
Cisco WSA in Transparent Mode	608
Configuring WCCP in a Cisco ASA to Redirect Web Traffic to a Cisco WSA	609
Configuring WCCP on a Cisco Switch	610
Configuring the Cisco WSA to Accept WCCP Redirection	612
Traffic Redirection with Policy-Based Routing	612
Cisco WSA Security Services	613
Deploying Web Proxy IP Spoofing	614
Configuring Policies in the Cisco WSA	615
Cisco WSA Reports	617
Cisco ESA	619
Reviewing a Few Email Concepts	619
Cisco ESA Deployment	620

	Cisco ESA Listeners	621
	SenderBase	622
	The Recipient Access Table (RAT)	622
	Cisco ESA Data Loss Prevention	622
	SMTP Authentication and Encryption	623
	Domain Keys Identified Mail (DKIM)	623
	Cisco Content Security Management Appliance (SMA)	624
	Exam Preparation Tasks	629
	Review All Key Topics	629
	Define Key Terms	630
	Review Questions	630
Chapter 11	Endpoint Protection and Detection	634
	“Do I Know This Already?” Quiz	634
	Foundation Topics	636
	Introduction to Endpoint Protection and Detection	636
	Endpoint Threat Detection and Response (ETDR) and Endpoint Detection and Response (EDR)	637
	Cisco AMP for Endpoints	638
	Outbreak Control	639
	IP Blacklists and Whitelists	643
	AMP for Endpoints Application Control	644
	Exclusion Sets	645
	AMP for Endpoints Connectors	648
	AMP for Endpoints Policies	648
	AnyConnect AMP Enabler	650
	AMP for Endpoints Engines	650
	AMP for Endpoints Reporting	651
	Cisco Threat Response	654
	Exam Preparation Tasks	655
	Review All Key Topics	655
	Define Key Terms	655
	Review Questions	656
Chapter 12	Final Preparation	658
	Hands-on Activities	658
	Suggested Plan for Final Review and Study	658
	Summary	659

Glossary of Key Terms 660

Appendix A Answers to the “Do I Know This Already?”
Quizzes and Q&A Sections 678

Appendix B CCNP Security Core SCOR (350-701) Exam Updates 686

Index 688

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.

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.



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.

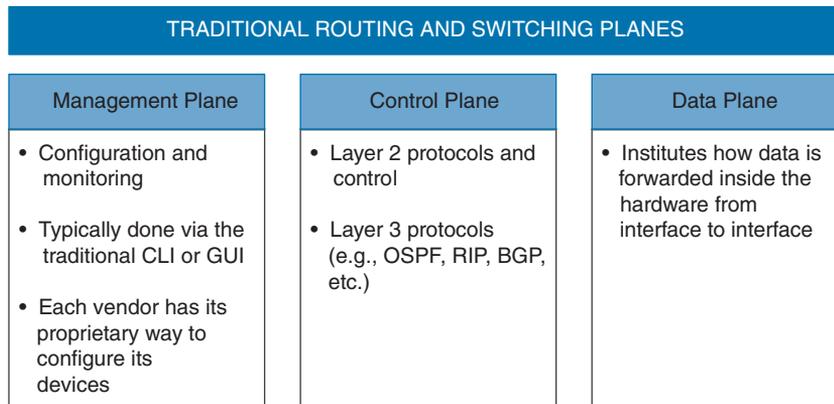


Figure 3-1 *The Management, Control, and Data Planes*

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.

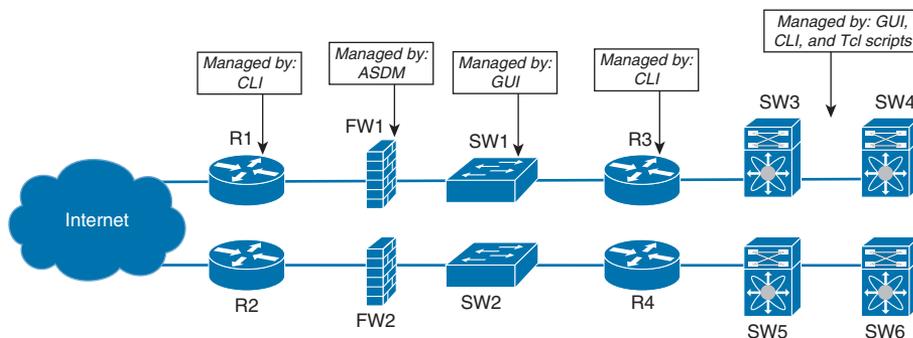


Figure 3-2 *Traditional Network Management Solutions*

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.

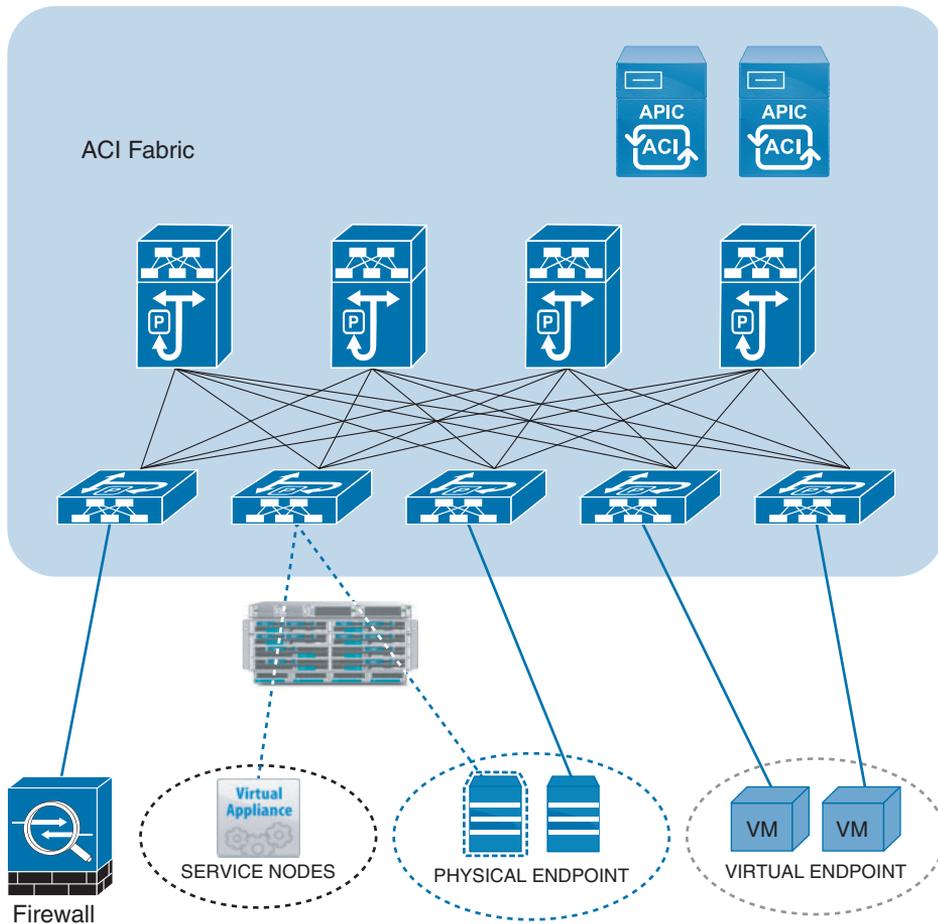


Figure 3-3 Cisco APIC Configuration and Policy Management

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.

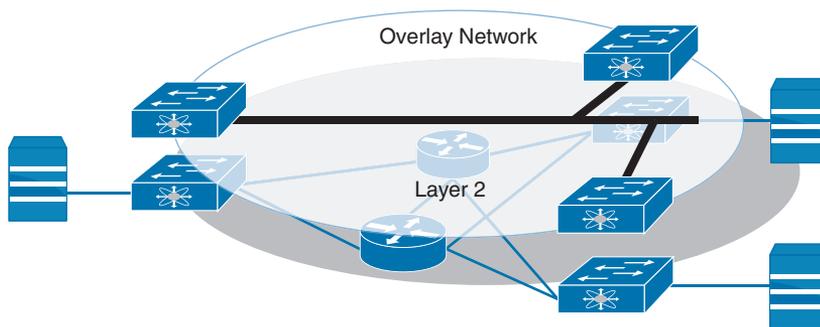


Figure 3-4 *Overlay Network Providing Layer 2 Capabilities*

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

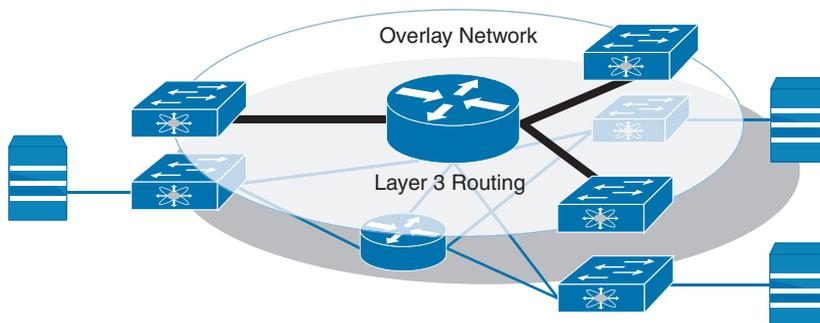


Figure 3-5 *Overlay Network Providing Layer 3 Routing Capabilities*

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

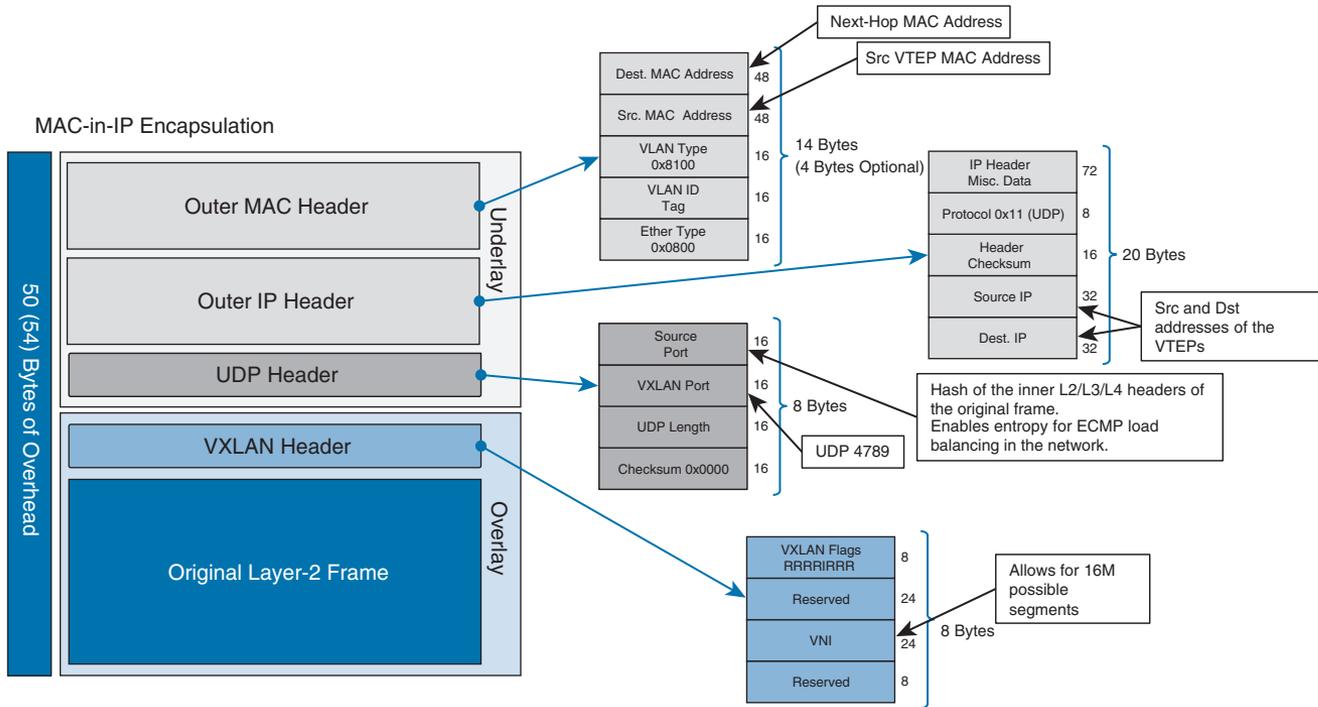


Figure 3-6 VXLAN Frame Format

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.

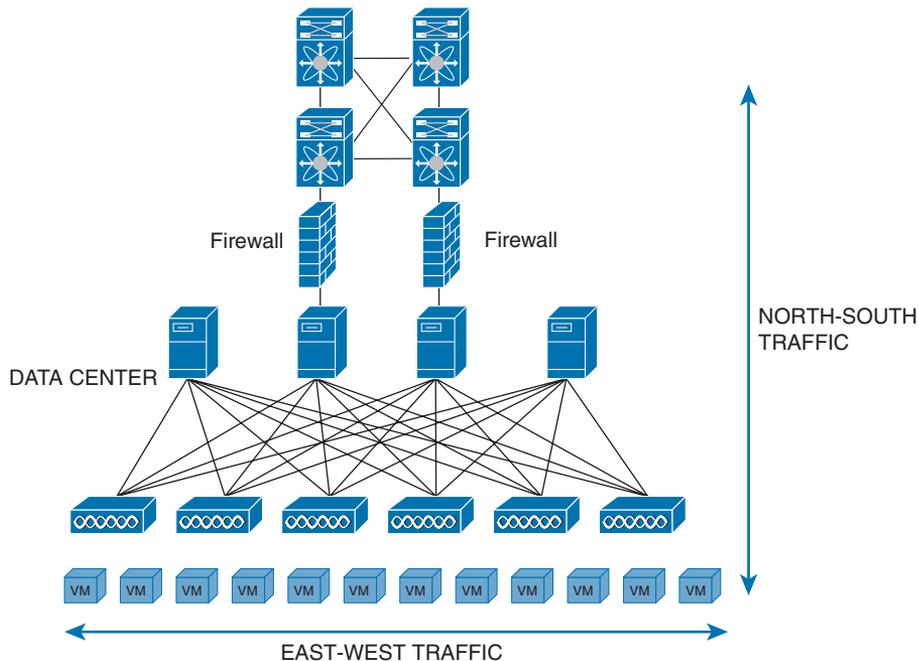


Figure 3-7 “East-West” and “North-South” Traffic

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).

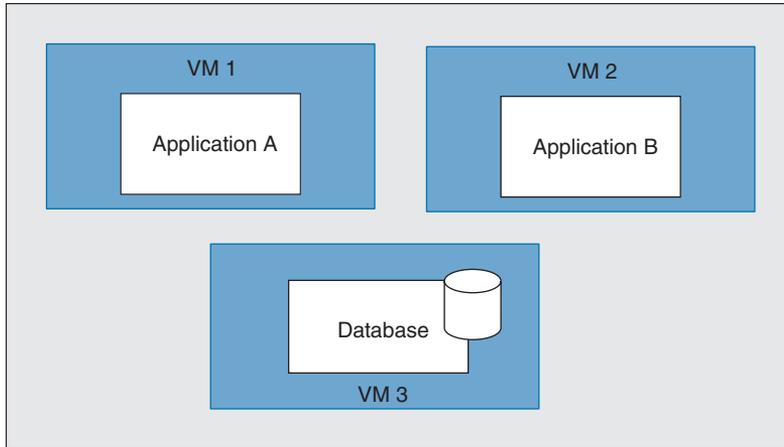


Figure 3-8 *Applications in VMs*

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).

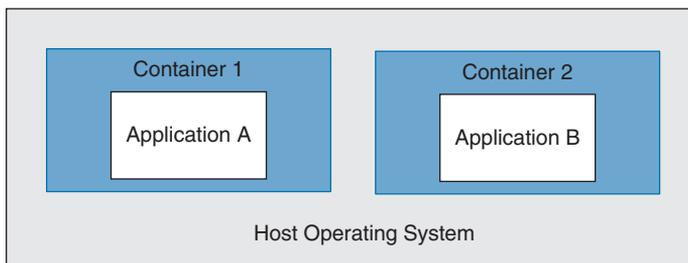


Figure 3-9 *Applications in Containers*

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.



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.

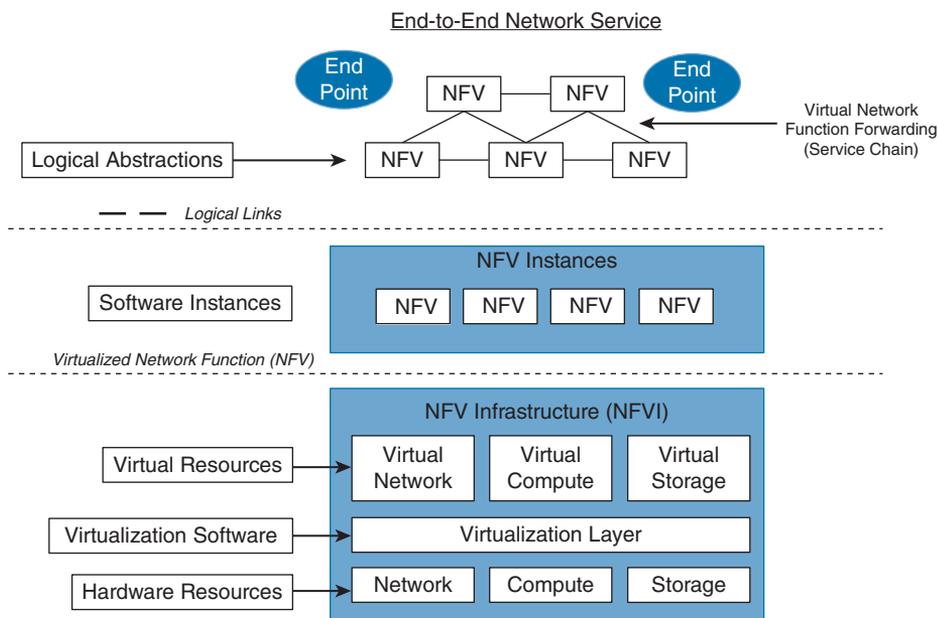


Figure 3-10 *NFV Architecture*

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.

NFV Orchestrator	VNF Manager	Virtualized Infrastructure Manager (VIM)
<ul style="list-style-type: none"> • On-boards (orchestrates) new network services (NS) and virtual network function (VNF) packages. • The NFV Orchestrator is also responsible for the lifecycle management; global resource management; validation and authorization of network functions virtualization infrastructure (NFVI) resource requests. 	<ul style="list-style-type: none"> • Oversees lifecycle management of VNF instances. • Coordinates configuration and event reporting between NFV infrastructure (NFVI) and Element/Network Management Systems. 	<ul style="list-style-type: none"> • Controls and manages the NFVI compute, storage, and network resources.

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>.

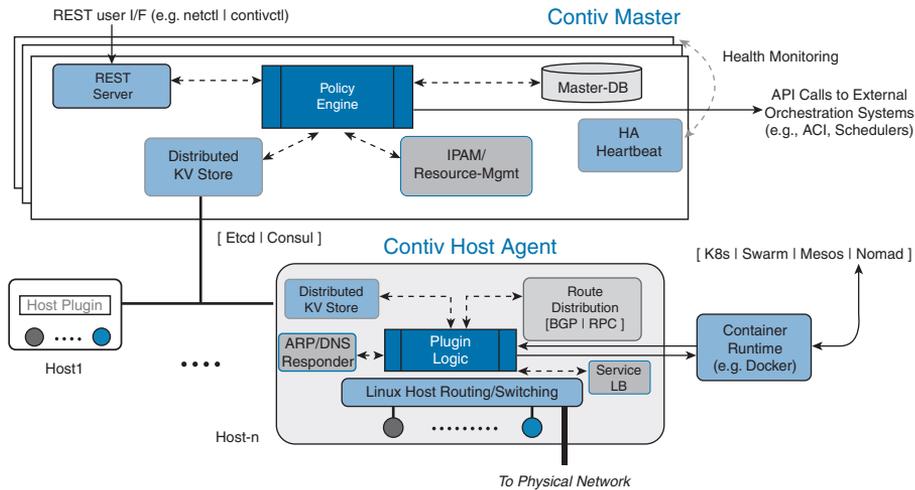


Figure 3-12 *Contiv Netmaster and Netplugin (Contiv Host Agent) Components*

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.

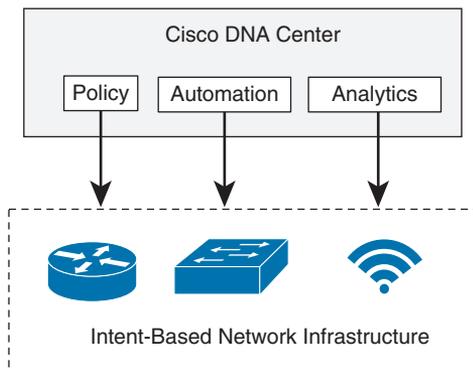


Figure 3-13 *Cisco DNA High-Level Architecture*

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.

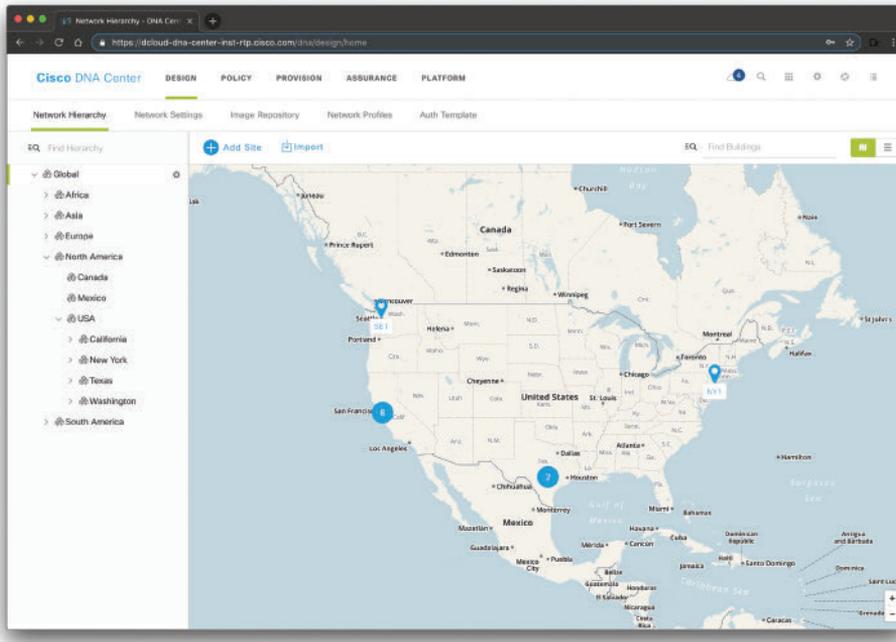


Figure 3-14 Cisco DNA Center Network Hierarchy Dashboard

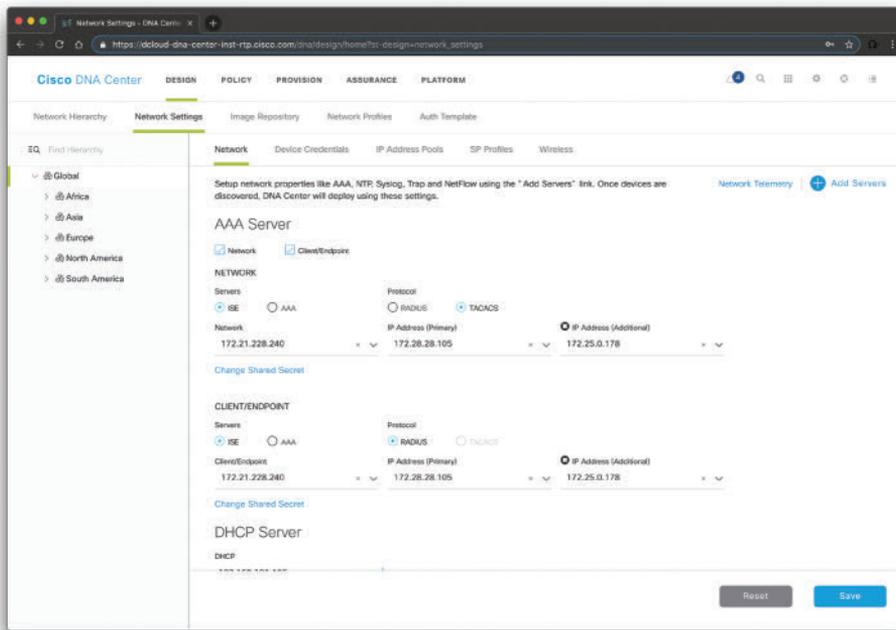


Figure 3-15 Cisco DNA Center Integration with Cisco ISE for AAA Services

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.

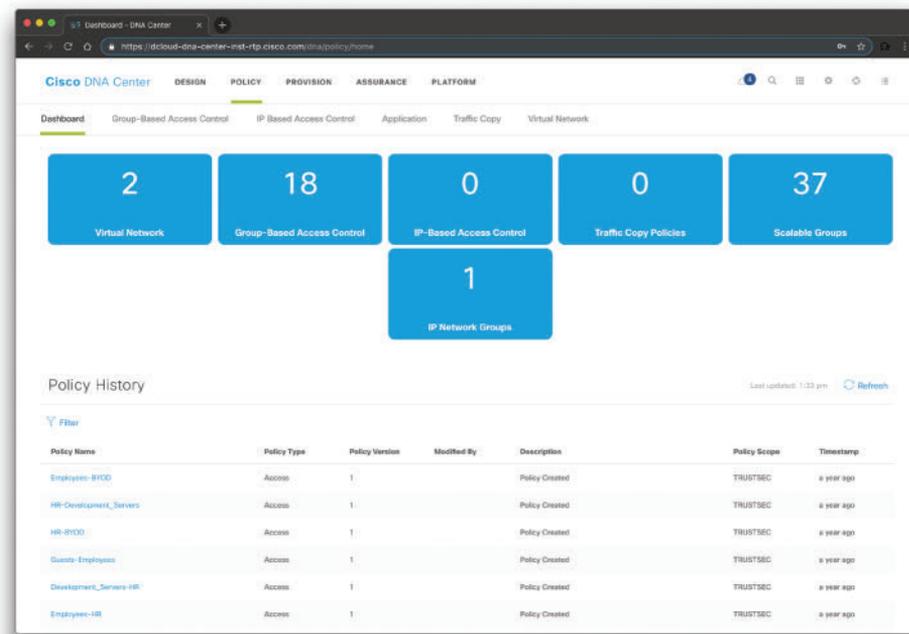


Figure 3-16 Cisco DNA Center Policy Dashboard

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.

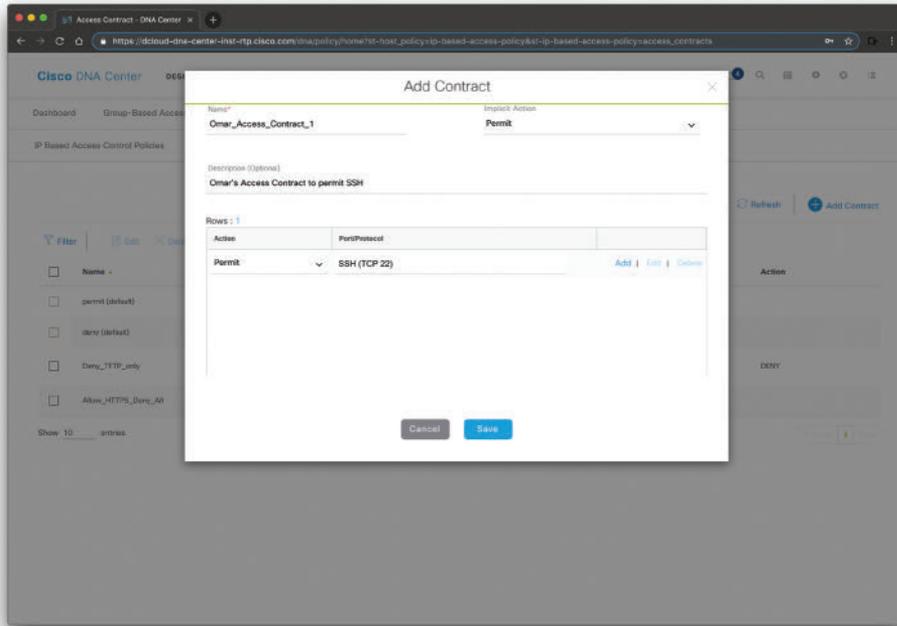


Figure 3-17 Adding a Cisco DNA Center Contract

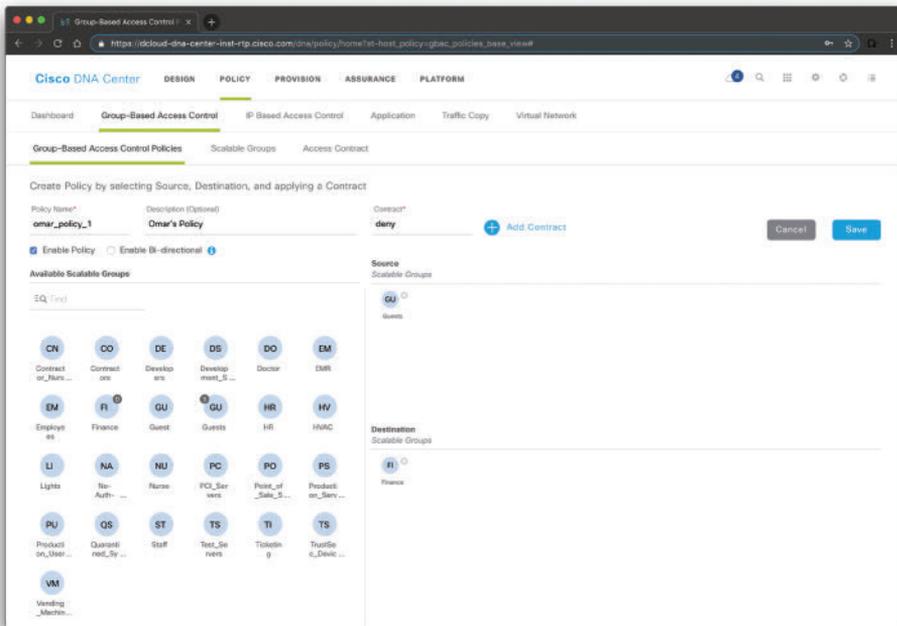


Figure 3-18 Adding a Cisco DNA Center Group-Based Access Control Policy

3

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.

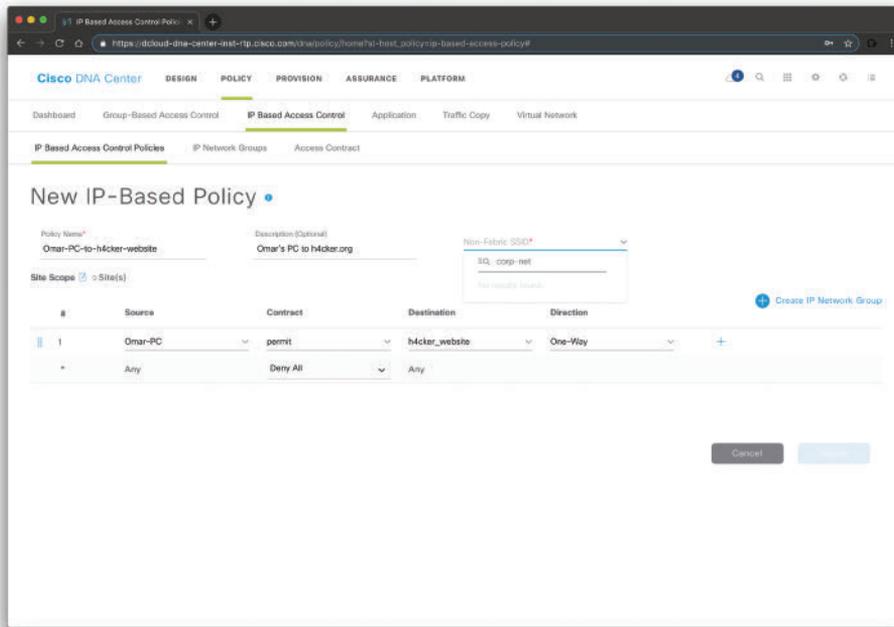


Figure 3-19 Adding a Cisco DNA Center IP-Based Access Control Policy

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.

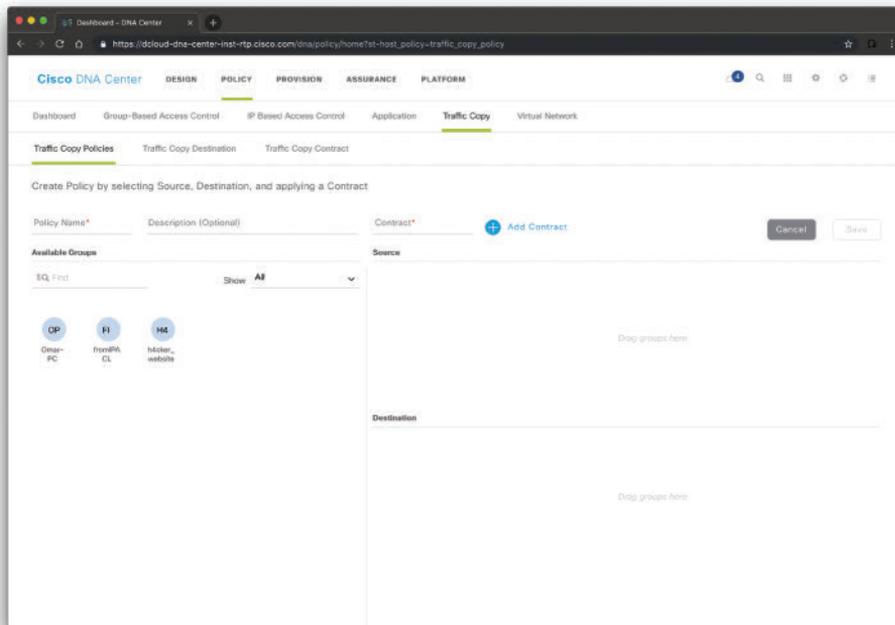


Figure 3-20 Adding a Traffic Copy Policy

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.

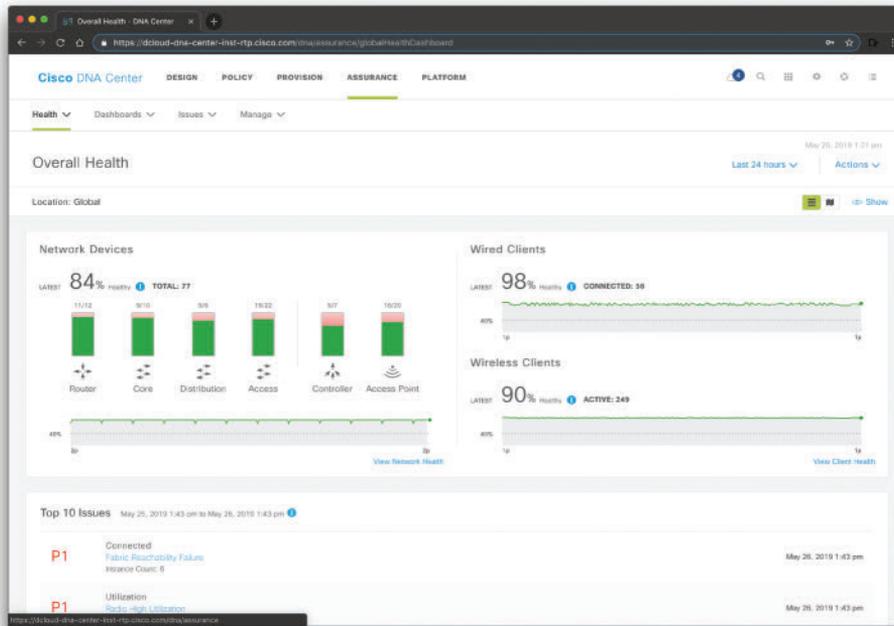


Figure 3-21 *The Cisco DNA Center Assurance Overall Health Dashboard*

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.

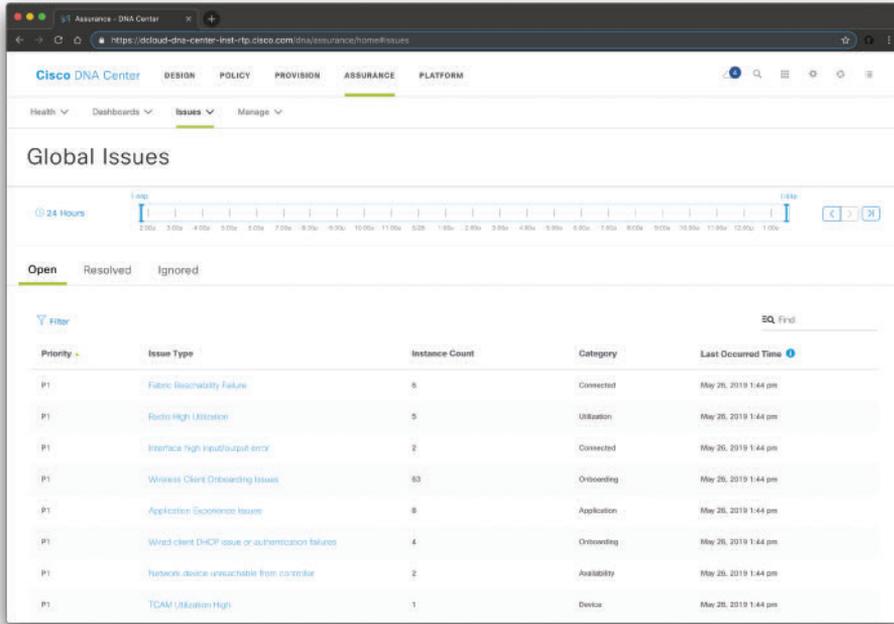


Figure 3-22 The Cisco DNA Center Assurance Global Issues Dashboard

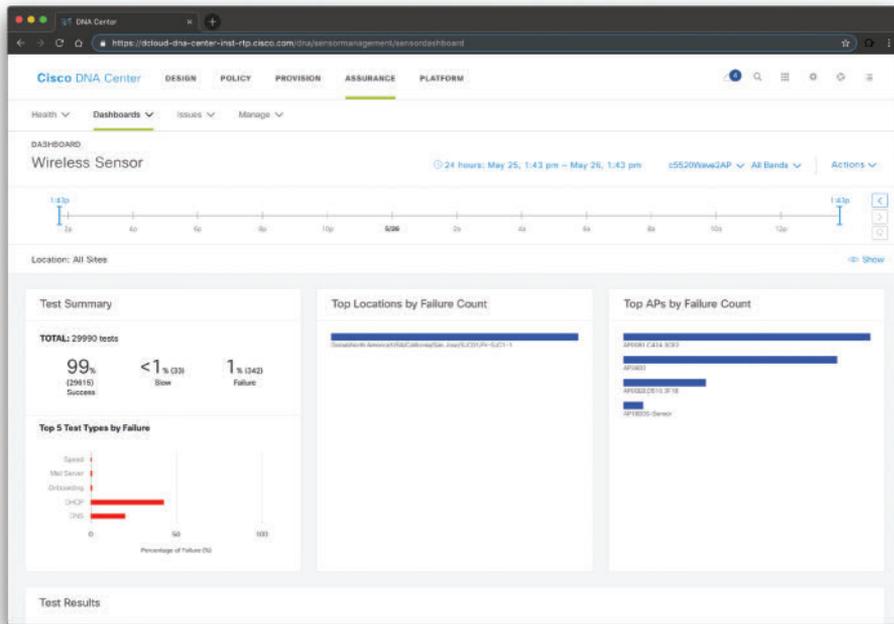


Figure 3-23 The Cisco DNA Center Assurance Wireless Sensor Dashboard

**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.

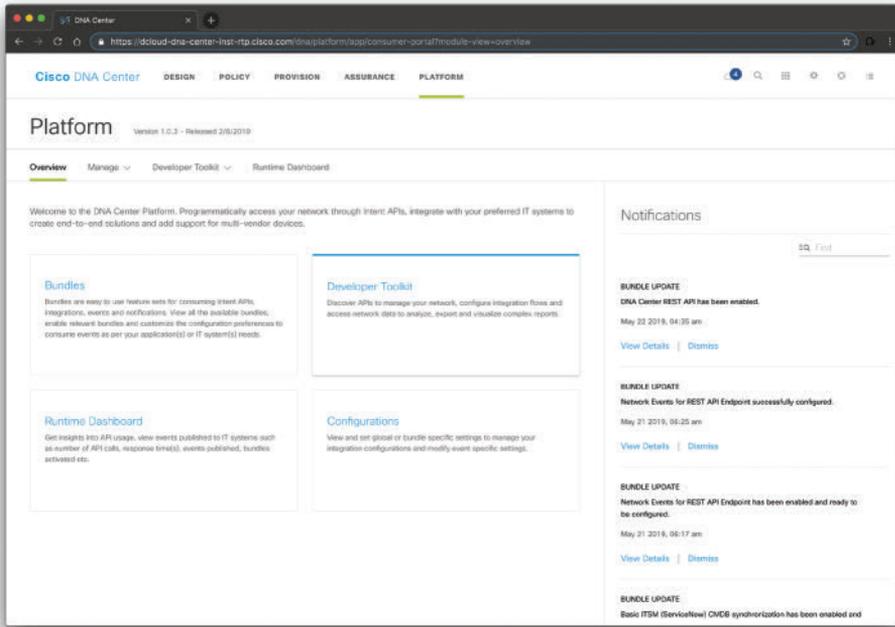
You can view information about all the Cisco DNA Center APIs by clicking the **Platform** tab and navigating to **Developer Toolkit > APIs**, as shown in Figure 3-24.

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>.



3

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

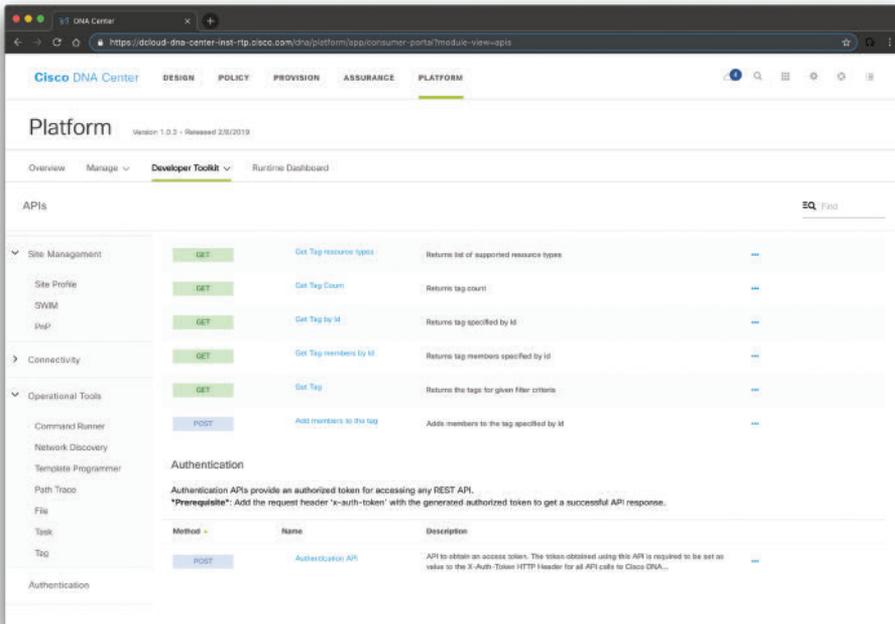


Figure 3-25 API Developer Toolkit Documentation



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.

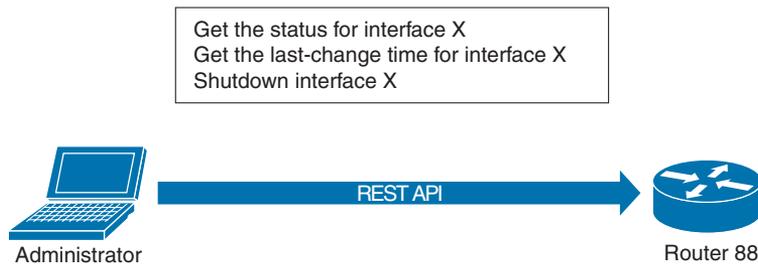


Figure 3-26 Using Network Infrastructure Device APIs

**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.

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>.



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>



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.

```
omar@omar_server_1]~$ curl -s https://deckofcardsapi.com/api/deck/new/shuffle?deck_count=1 | python -m json.tool
{
  "remaining": 52,
  "shuffled": true,
  "deck_id": "wkc12q20frlh",
  "success": true
}
omar@omar_server_1]~$
```

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

```
omar@omar_server_1]~$ curl -s https://deckofcardsapi.com/api/deck/new/shuffle?deck_count=1 | python -m json.tool
{
  "remaining": 52,
  "shuffled": true,
  "deck_id": "wkc12q20frlh",
  "success": true
}
omar@omar_server_1]~$ curl -s https://deckofcardsapi.com/api/deck/wkc12q20frlh/draw/ | python -m json.tool
{
  "remaining": 51,
  "cards": [
    {
      "code": "9S",
      "suit": "SPADES",
      "value": "9",
      "images": {
        "png": "https://deckofcardsapi.com/static/img/9S.png",
        "svg": "https://deckofcardsapi.com/static/img/9S.svg"
      },
      "image": "https://deckofcardsapi.com/static/img/9S.png"
    }
  ],
  "deck_id": "wkc12q20frlh",
  "success": true
}
omar@omar_server_1]~$
```

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

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.



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.



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.

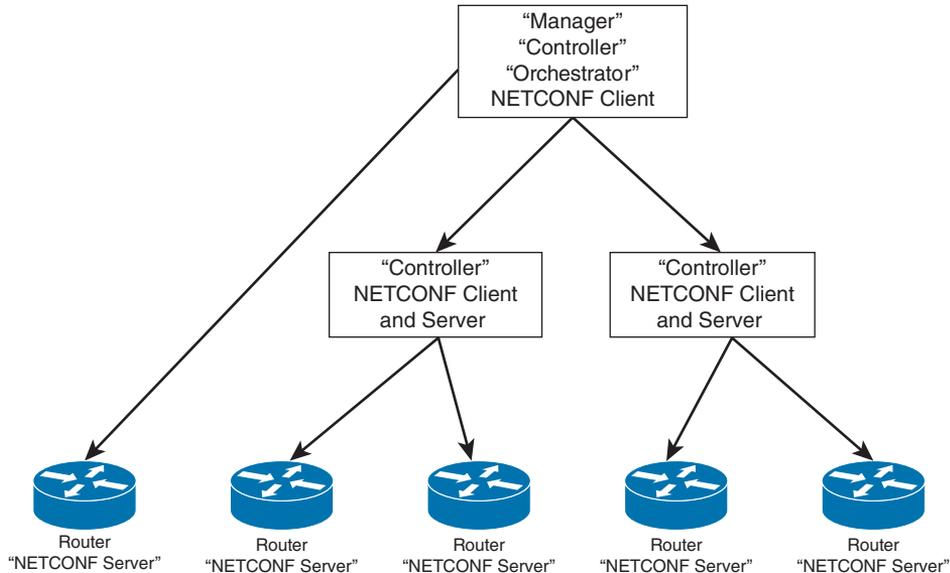


Figure 3-32 NETCONF Clients, Servers, and Controllers

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.

```

omar@omar_server_1:~$ ssh root@ios-xe-mgmt.cisco.com -p 10000 -s netconf
root@ios-xe-mgmt.cisco.com's password:
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability urn:ietf:params:netconf:base:1.0/>
    <capability urn:ietf:params:netconf:base:1.1/>
    <capability urn:ietf:params:netconf:capability:writable-running:1.0/>
    <capability urn:ietf:params:netconf:capability:xpath:1.0/>
    <capability urn:ietf:params:netconf:capability:validate:1.0/>
    <capability urn:ietf:params:netconf:capability:validate:1.1/>
    <capability urn:ietf:params:netconf:capability:rollback-on-error:1.0/>
    <capability urn:ietf:params:netconf:capability:notification:1.0/>
    <capability urn:ietf:params:netconf:capability:interleave:1.0/>
    <capability urn:ietf:params:netconf:capability:with-defaults:1.0?basic-mode=explicit&also-supported=report-all-tagged/>
    <capability urn:ietf:params:netconf:capability:yang-library:1.0?revision=2016-06-21&module-set-id=88c694c75e847aba17e8ab19254ad090/>
    <capability http://tail-f.com/ns/netconf/actions/1.0/>
    <capability http://tail-f.com/ns/netconf/extensions/>
    <capability http://cisco.com/ns/cisco-xe-ietf-ip-deviation?module=cisco-xe-ietf-ip-deviation&revision=2016-08-10/>
    <capability http://cisco.com/ns/cisco-xe-ietf-ipv4-unicast-routing-deviation?module=cisco-xe-ietf-ipv4-unicast-routing-deviation&revision=2015-09-11/>
    <capability http://cisco.com/ns/cisco-xe-ietf-ipv6-unicast-routing-deviation?module=cisco-xe-ietf-ipv6-unicast-routing-deviation&revision=2015-09-11/>
    <capability http://cisco.com/ns/cisco-xe-ietf-ospf-deviation?module=cisco-xe-ietf-ospf-deviation&revision=2018-02-09/>
    <capability http://cisco.com/ns/cisco-xe-ietf-routing-deviation?module=cisco-xe-ietf-routing-deviation&revision=2016-07-09/>
    <capability http://cisco.com/ns/cisco-xe-openconfig-acl-deviation?module=cisco-xe-openconfig-acl-deviation&revision=2017-08-25/>
    <capability http://cisco.com/ns/mpls-static/devs?module=common-mpls-static-devs&revision=2015-09-11/>
    <capability http://cisco.com/ns/nvo/devs?module=nvo-devs&revision=2015-09-11/>
  </capabilities>
</hello>

```

Figure 3-33 Using the NETCONF SSH Subsystem

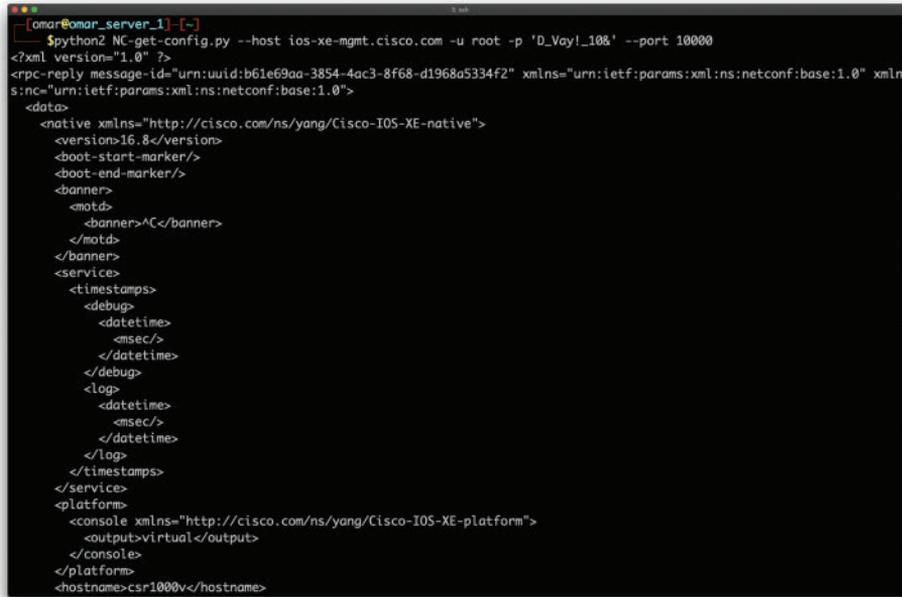
TIP DevNet has several sandboxes where you can practice these concepts and more at <https://devnetsandbox.cisco.com>.

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_Vayl_100!' --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" xmlns:s="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>
              <mssec/>
            </datetime>
          </debug>
          <log>
            <datetime>
              <mssec/>
            </datetime>
          </log>
        </timestamps>
      </service>
      <platform>
        <console xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-platform">
          <output>virtual</output>
        </console>
      </platform>
      <hostname>csr1000v</hostname>
    </native>
  </data>
</rpc-reply>

```

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)

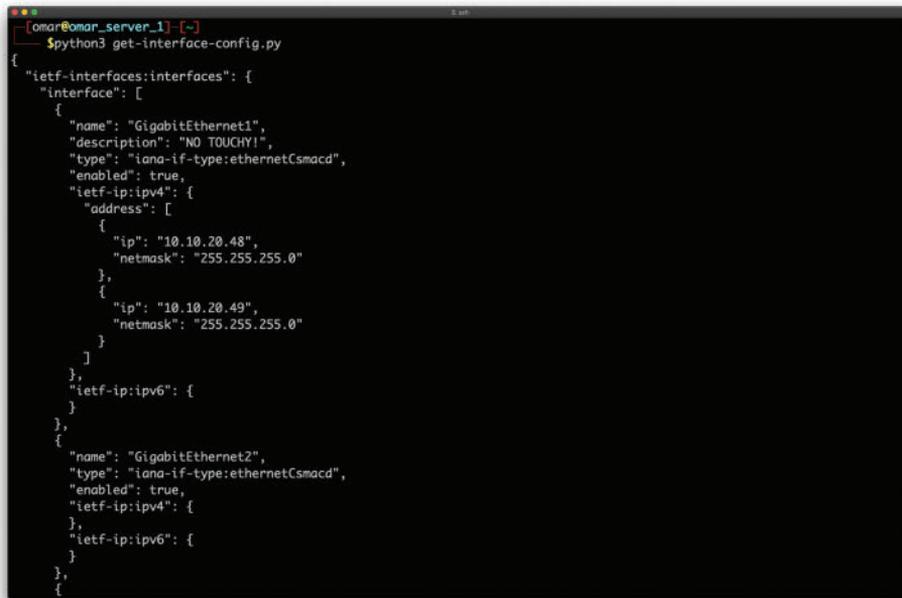
    # RESTCONF media types for REST API headers
    headers = {'Content-Type': 'application/yang-data+json',
              'Accept': 'application/yang-data+json'}

    # 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`).



```

[omar@omar_server_1]~$ python3 get-interface-config.py
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "GigabitEthernet1",
        "description": "NO TOUCHY!",
        "type": "iana-if-type:ethernetCsmacd",
        "enabled": true,
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "10.10.20.48",
              "netmask": "255.255.255.0"
            },
            {
              "ip": "10.10.20.49",
              "netmask": "255.255.255.0"
            }
          ]
        },
        "ietf-ip:ipv6": {}
      },
      {
        "name": "GigabitEthernet2",
        "type": "iana-if-type:ethernetCsmacd",
        "enabled": true,
        "ietf-ip:ipv4": {},
        "ietf-ip:ipv6": {}
      }
    ]
  }
}

```

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.



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

1-to-1 signatures, 454
6LoWPAN (IPv6 over Wireless Personal Area Networks), 54
802.1D. *See* STP
802.1X, 178–180, 187, 324
 active policy enforcement, 295–298
 authentication
 configurations, 205–211
 failures, 203
 C3PL, 204–205
 monitor mode deployments, 294–295
 Multi-Auth mode, 203
 Open Authentication, 204
 port security, 203

A

AAA (Authentication, Authorization, Accounting), 154–155
 802.1X, 178–180
 aaa-new model command, 358
 accounting, 169–170
 ACL, 170–172
 ACM, 171–172
 authentication, 155–156, 158–159
 by authentication, 157–158
 BeyondCorp, 163
 centralized (linked) identities, 165–166
 Duo Security, 159–163
 EAP, 179
 EAPoL, 179
 federated identities, 165–166
 Flex-Auth, 203
 guest access, 188
 ISE authorization rules, 187–188
 by knowledge, 155–156
 Multi-Auth mode (802.1X), 203
 multifactor authentication, 159, 160–161, 166
 Open Authentication, 204
 passwords, 156–157
 by possession, 157–158
 RADIUS, 173–176, 179, 187
 SAML, 159, 165
 single-factor authentication, 159
 web authentication, 187–188
 zero-trust, 161–167
 authorization, 167
 ABAC, 169
 ACL, 167
 CoA, 193–196
 DAC, 168
 implicit deny, 168
 MAB, 188, 203
 MAC, 168
 need to know, 168
 RBAC, 168–169
 rule-based access control, 169
 security labels, 167

- capability tables, 171
- CLI, troubleshooting AAA for Cisco routers, 353–356
- content-dependent access control, 172
- context-dependent access control, 172
- dACL, 181
- Diameter, 176–179
- digital certificates, 100
- firewalls, 180
- infrastructure access controls, 170
- IPv4, 373
- IPv6, 373
- management plane (NFP), 335, 339
- method lists, 343, 349–353
- network ACL, 180
- principle of least privilege, 155
- RADIUS, 173–176, 179, 202–205
- separation of duties, 155
- SGACL, 181
- TACACS+, 174–176
 - access configuration*, 196–199, 200–202
 - debugging*, 199–200
- test aaa command, 356
- user authentication, 342, 349–353
- VLAN ACL, 181
- ABAC (Attribute-Based Access Control), 169**
- access**
 - cloud computing, 49
 - Duo Access Gateways, 160
 - group-based access control policy (DNA), 124
 - ip-based access control policy (DNA), 124
 - physical access, Trojans, 21
 - port-based access lists, IPv6, 377
 - router access authentication, 342–343
 - SOAP, 136, 137
 - traffic copy policy (DNA), 127
 - unauthorized access, IPv4/IPv6, 374
 - zone access, IPv4/IPv6, 373
- access control management, 154, 427**
 - ABAC, 169
 - access control mechanisms, 170–172
 - ACE, 427, 432, 435
 - ACL, 167, 170–172, 435–436
 - ASA and*, 427–433
 - characteristics of*, 429–430
 - EtherType ACL*, 431
 - extended ACL*, 430–431
 - global ACL*, 429
 - HTTP traffic*, 432–433
 - interface ACL*, 429
 - network ACL*, 180
 - object grouping*, 435–436
 - SMTP traffic*, 432–433
 - standard ACL*, 430
 - time-based ACL*, 436–437
 - types of*, 424–431
 - Webtype ACL*, 431, 523–524
- ACM, 171–172
- capability tables, 171
- content-dependent access control, 172
- context-dependent access control, 172
- DAC, 47, 168
- dACL, 181
- factors, 47
- FTD and access control policies, 443–445
- guest access, 188
- infrastructure access controls, 170
- MAC, 47, 168
- network ACL, 180
- privileges, 45–46

- process of, 46
- RADIUS, authentication configuration, 202–205
- RBAC, 47, 168–169
- rule-based access control, 169
- SGACL, 181
- TACACS+, 174–176
 - access configuration*, 196–199, 200–202
 - debugging*, 199–200
- TrustSec, 190–192
 - ACI integration*, 298–301
 - active policy enforcement*, 295–298
 - monitor mode deployments*, 294–295
 - SGT*, 188
- VLAN ACL, 181
- access-group command, 435
- access-list command, 436
- accounting, 169–170
- ACE (Access Control Entries), 427, 432, 435
- ACI (Application Centric Infrastructure), 110–112
 - micro-segmentation, 289–290
 - TrustSec integration, 298–301
- ACK packets, 25
- AckCmd, 25
- acknowledgements (TCP), 25
- ACL (Access Control Lists), 167, 170–172, 181, 324, 435–436
 - ACE, 427, 432, 435
 - ASA and, 427–433
 - characteristics of, 429–430
 - CoPP, permitted ACL traffic, 381
 - dACL, 181
 - EtherType ACL, 431
 - extended ACL, 430–431
 - global ACL, global ACL, 429
 - HTTP traffic, 432–433
 - interface ACL, 429
 - IPv6, 377–378
 - logging, 380
 - network ACL, 180
 - object grouping, 435–436
 - SGACL, 181
 - SMTP traffic, 432–433
 - standard ACL, 430
 - time-based ACL, 436–437
 - types of, 424–431
 - VLAN ACL, 181
 - Webtype ACL, 431, 523–524
- ACM (Access Control Matrix), 171–172
- address proxying, IPS/IDS, 58
- address spoofing, IPS/IDS, 58
- ADM (Application Dependency Mapping), 594
- admin-context command, 415
- administrator views, 344
- advertising, spyware, 26
- AFL (American Fuzzy Lop), 573
- agile development methodology (cloud computing), 553–556, 559
- algorithms. *See also* ciphers
 - asymmetric algorithms, 83–84
 - cryptographic algorithms, 470
 - symmetric encryption algorithms, 82–83
 - thumbprint algorithms
 - digital certificates*, 98
 - root certificates*, 96
- all-nodes multicast addresses, IPv6, 368
- all-routers multicast addresses, 368

- AMP (Advanced Malware Protection), 452, 582
 - 1-to-1 signatures, 454
- AMP for Endpoints, 637, 638–639
 - AnyConnect AMP Enabler*, 650
 - Application Control*, 644–645
 - connectors*, 648
 - Ethos*, 650
 - exclusion sets*, 645–647
 - IP blacklists/whitelists*, 643–644
 - Outbreak Control*, 639–643
 - policies*, 648–649
 - reports*, 651–654
 - Spero*, 650
 - TETRA*, 650
- AMP for Networks, 452
- AnyConnect AMP Enabler, 650
- architecture of, 637
- cloud computing, 452–454
- Ethos, 454
- features of, 452
- hashes, 85
- IOC, 454, 455
- retrospection, 456–457
- Spero, 454
- Threat Grid, 452–453, 455–456
- amplification attacks, availability (CIA triad), 45
- annotations, STP, 318–320
- anomaly detection, NetFlow, 229–231
- antidetection routines, 18
- anycast addresses, IPv6, 368–369
- AnyConnect, 193
 - AMP Enabler, 650
 - posture assessments, 192–193
 - Secure Mobility, 478–479
 - DTLS*, 529
 - split tunneling*, 528–529
 - VPN*, 527–529
 - stealth AnyConnect, 193
- Apache Mesos, 560
- API (Application Programming Interface), 38
 - attacks, cloud computing, 51
 - DNAC API, 130, 132
 - documentation, 39
 - GraphQL, 38
 - IoT, 53
 - network programmability, 132–133, 136
 - GraphQL*, 137
 - network device API*, 139
 - REST*, 137–139
 - SOAP*, 136, 137
 - Swagger (OpenAPI)*, 137
 - WADL documents*, 137
 - WSDL documents*, 137
 - YANG models*, 139–140
 - northbound API, 118
 - OpenAPI, Swagger, 39
 - REST, 38
 - RESTful API, cloud computing, 53
 - SOAP, 38
 - southbound API, 118
 - Swagger, 39
 - WADL, 39
 - WSDL, 39
- APIC (Application Policy Infrastructure Controller), 110–112
- AppDynamics Cloud Monitoring, 590–593
- application layer
 - attacks, IPv4/IPv6, 373–374
 - tunneling, 25

applications

- access, SSL VPN, 524–525
- ADM, 594
- application sets, 127
- ASA application inspection, 433–434
- assurance tools/methods, 572–573
- DAST, 572–573
- impersonated mobile apps, Trojans, 21
- mapping, RFC 4594, 127
- network segmentation, 288–289
- policies (DNA), 126–127
- recognition, AVC, 241–242
- SAST, 572–573
- tracking with NetFlow, 270–271
- vulnerabilities, 9
- APT (Advanced Persistent Threats), 20**
- ARP (Address Resolution Protocol)**
 - cache poisoning, 330–331
 - DAI, 330–332
 - requests, IPv6, 380
 - spoofing
 - attacks*, 330
 - data plane (NFP)*, 338
- ASA (Adaptive Security Appliance), 396**
 - ACL, 427–433, 435–436
 - object grouping*, 435–436
 - time-based ACL*, 436–437
 - application inspection, 433–434
 - ASAv, 396
 - assigning
 - inside/outside networks*, 412
 - interface addresses*, 412
 - IP addresses*, 412
 - security levels*, 412, 432
 - CDO, 408–410
 - DHCP and, 412
 - failover (high-availability) mode, 423–425
 - FirePOWER module, 396–397, 401–404
 - firewalls, 187
 - ICMP filtering, 437
 - internal to global address translation, 438
 - MMTF, 416
 - MPF, 433
 - NAT, 437–438, 443
 - auto-NAT*, 443
 - dynamic NAT*, 438, 441–442
 - identity NAT*, 442
 - manual NAT*, 443
 - policy NAT*, 442
 - static NAT*, 438, 441
 - support*, 396
 - TCP Intercept*, 443
 - one-to-one address mapping, 438
 - PAT, 440
 - dynamic PAT*, 442
 - policy PAT*, 442
 - static PAT*, 441
 - port redirection, 441
 - remote-access VPN configurations, 511–512
 - attributes*, 518
 - client-based remote access SSL VPN*, 524–526
 - clientless remote access SSL VPN*, 514–515
 - design considerations*, 515–516
 - group policy*, 513
 - IP pools*, 513
 - IPsec*, 512–514
 - NAT exemptions*, 514
 - policy inheritance model*, 518
 - tunnel groups*, 513–514

- routed firewalls, 413
- site-to-site VPN configurations, 502–503
 - bypass NAT*, 508–509
 - crypto maps*, 506–508
 - fragmentation*, 510–511
 - IPsec policies*, 505–506
 - ISAKMP*, 503–504
 - management access*, 510
 - NAT-T*, 510
 - OSPF over IPsec*, 509
 - PFS*, 509
 - traffic-filtering*, 503–508
 - tunnel default gateways*, 510
 - tunnel groups*, 504–505
- SMTF, 414–416
- TCP Intercept, 443
- traffic-filtering, 396–397
 - to-the-box traffic filtering*, 434–435
 - through-the-box traffic filtering*, 431
- transparent firewalls, 413–414
 - MMTF*, 416
 - SMTF*, 414–416
- WCCP
 - ASA configurations*, 609–610, 612
 - web traffic redirection to WSA*, 609–610, 612
- ASCII armoring, 39
- ASDM, site-to-site VPN ASA firewall configurations, 505
- ASLR (Address Space Layout Randomization), 39
- assets, defined, 12
- Assurance solution, DNAC, 128–129
- asymmetric algorithms, 83–84
- AsyncOS (Async Operating System), 604
- attachments (email), Trojans, 21
- auditing, cloud computing, 49
- authentication, 155–156
 - access control management, 47
 - by authentication, 157–158
 - authentication attacks, cloud computing, 51
 - BeyondCorp, 163
 - CA, 98–99
 - centralized (linked) identities, 165–166
 - by characteristic, 158–159
 - Duo Security, 159–163
 - EAP, 179
 - EAPoL, 179
 - federated identities, 165–166
 - Flex-Auth, 203
 - guest access, 188
 - HMAC, 86
 - keychain authentication (BGP), 387
 - by knowledge, 155–156
 - management plane (NFP), 339
 - MD5
 - BGP*, 386–387
 - EIGRP authentication*, 384–385
 - OSPF authentication*, 383–384
 - RIP*, 385–386
 - memory cards, 158
 - method lists, 343, 349–353
 - Multi-Auth mode (802.1X), 203
 - multifactor authentication, 15–161, 166, 341
 - multilayer authentication, 47
 - NTPv3 authentication keys, 363–364
 - Open Authentication, 204
 - OTP, 157–158
 - out-of-band authentication, 47, 158
 - passwords, 156–158

- by possession, 157–158
- RADIUS, 173–176, 179, 187, 196–199
- router access authentication, 342–343
- routing update authentication
 - BGP*, 386–387
 - RIP*, 385–386
- SAML, 159, 165
- single-factor authentication, 47, 159, 341
- smartcards, 158
- SMTP, ESA, 623
- SSL VPN
 - application access*, 524–525
 - enabling*, 522–523
 - user authentication*, 520–522
 - Webtype ACL*, 523–524
- TACACS+, 196–199, 200–202
- user authentication, 342, 349–353
- vulnerabilities, 32–33
 - credential brute force attacks*, 33–34
 - cryptographic algorithms*, 33
 - default credentials*, 34
 - insecure direct object reference vulnerabilities*, 35
 - password cracking*, 33–34
 - rainbow tables*, 33–34
 - session hijacking*, 34
 - WEP*, 34
- web authentication, 187–188
- zero-trust, 161–167

authorization

- ABAC, 169
- ACL, 167
- CoA, 193–196
- DAC, 168
- implicit deny, 168
- ISE authorization rules, 187–188

- MAB, 188, 203
- MAC, 168
- need to know, 168
- object capability, 167
- RBAC, 168–169
- rule-based access control, 169
- security labels, 167
- auto secure command-line utility**, NFP, 334
- autoconfiguration**, IPv6, 376
- auto-NAT**, 443

- availability (CIA triad)**, 43–44
 - amplification attacks, 45
 - buffer overflows, 45
 - DDoS attacks, 45
 - direct DoS attacks, 44
 - DoS attacks, 40–42
 - reflected DoS attacks, 45
- availability checks, management plane (NFP)**, 341
- AVC (Application Visibility and Control)**, 241
 - application recognition, 241–242
 - metrics collection/exportation, 242
- AWS (Amazon Web Services)**
 - CDO, 409–410
 - Lambda, 559

B

- backdoors, Trojans**, 19
- Bad Rabbit**, 23
- in-band SQL injection**, 32
- bandwidth**
 - low-bandwidth attacks, IPS/IDS, 58
 - managing, data plane (NFP), 338
- BCP (Business Continuity Plans), cloud computing**, 50

BGP (Border Gateway Protocol)

- keychain authentication, 387
- MD5 authentication, 386–387
- routing update authentication, 386–387

BinText, 27**biometric security, 158–159****BIOS infections, 16****black hat hackers, 14****blacklists/whitelists (IP), 643–644****BLE (Bluetooth Low Energy), IoT, 53****blind (inferential) SQL injection, 32****block ciphers, 82****Blueprints (exams), 658****Bluetooth**

- BLE, IoT, 53
- Bluetooth Smart, IoT, 53
- malware, 17

bogus IPv6 addresses, filtering, 376**bootsets (secure), creating, 364–365****bot hosts, 230****botnets, 45, 230****BPDU Guard, 324–325****breaches (data)**

- examples of, 156
- IOC, 454, 455

browsers

- extensions, 21
- man-in-the-browser attacks, 34
- Trojans, 21
- vulnerabilities, 21
- XSS testing, 37

brute force credential attacks, 33–34**buffer overflows, 39, 45****bugs in code, IPv6, 376****business continuity. *See* BCP****BVI (Bridge Virtual Interface) and FTD, 417–419****C****C3PL (Cisco Common Classification Policy Language), 204–205****CA (Certificate Authority), 87–88 93–94**

- authentication, 98–99
- commercial CA, 94
- cross-certifying CA, 102
- digital certificates, 94, 97–98
- enrollment, 98–99
- hierarchical CA, 101–102
- identity certificates, 94, 96–97
- root certificates, 95–96
- single root CA, 101
- subordinate CA, 101
- system root CA certificates, 88–89

cables (console), management plane (NFP), 339**caches**

- NetFlow, 228–229
- poisoning (ARP), 330–331

CAM overflow attacks, data plane (NFP), 338**capability tables, 171****CASB (Cloud Access Security Broker), 584****CASE (Context Adaptive Scanning Engine), 582****cat Linux command, 84****catastrophic damage, threats, 12****CD (Continuous Delivery), CI/CD pipelines, 558–559, 572****CDO (Cisco Defense Orchestrator), 408–410****CDP (Cisco Discovery Protocol)**

- disabling, 327–328
- Layer 2 security, 327–328

CEF (Cisco Express Forwarding), 337

- cellular connections, IoT, 54
- centralized (linked) identities, 165–166
- Centri Firewall, 396
- CER (Crossover Error Rates), 159
- CERT (Computer Emergency Response Teams), 66, 71–72
- chain of custody, digital forensics, 59
- characteristics, authorization by, 158–159
- CI (Continuous Integration), CI/CD pipelines, 558–559, 572
- CIA triad, 40
 - availability, 43–44
 - amplification attacks*, 45
 - buffer overflows*, 45
 - DDoS attacks*, 45
 - direct DoS attacks*, 44
 - DoS attacks*, 40–42
 - reflected DoS attacks*, 45
 - confidentiality, 40–43
 - integrity, 42–43
- ciphers. *See also* algorithms
 - block ciphers, 82
 - ciphertext streams, 82
 - defined, 80
 - digit streams, 82
 - polyalphabetic method, 81
 - stream ciphers, 82
 - substitution method, 81
 - transposition method, 81
- classifying data, cloud computing, 49
- CLI (Command-Line Interface), troubleshooting AAA for Cisco routers, 353–356
- client-based remote access SSL VPN, 522–523
 - configurations, 525
 - group policy, 525–526
 - tunnel policies, 525–526
- clientless SSL VPN
 - application access, 524–525
 - enabling, 522–523
 - remote access SSL VPN, 514–515
 - user authentication, 520–522
 - Webtype ACL, 523–524
- closed-loop functioning, IoT, 51
- cloud computing, 47–48, 50
 - access, 49
 - advantages of, 47
 - agile development methodology, 553–556, 559
 - AMP, 452–454
 - Apache Mesos, 560
 - API attacks, 51
 - AppDynamics Cloud Monitoring, 590–593
 - attacks, 50–51
 - auditing, 49
 - authentication attacks, 51
 - AWS Lambda, 559
 - BCP, 50
 - CASB, 584
 - characteristics of, 48, 551
 - CI/CD pipelines, 558–559, 572
 - Cloudlock, 584–589
 - community clouds, 48, 552
 - containers, 561
 - Apache Mesos*, 560
 - Docker Swarm*, 561
 - images*, 561–565
 - Katacoda container deployments*, 563
 - Kubernetes*, 559
 - Nomad*, 560
 - OCI*, 561
 - orchestration*, 559–561
 - registries*, 561

- Contiv, 571
- contracts, ending, 50
- cryptographic attacks, 50
- CSP
 - customer/provider cloud security responsibilities, 573–575*
 - penetration testing, 575–577*
 - questions to ask, 575–577*
- customer/provider security responsibilities, 573–575
- data classification systems, 49
- data separation, 49
- DDoS attacks, 50
- deployment models, 48
- DevOps, 552, 556–557
- DevSecOps, 571
 - assurance tools/methods, 572–573*
 - CI/CD pipelines, 572*
 - OWASP Proactive Controls, 571–572*
 - tutorials, 571*
- DNS attacks, 50
- Docker
 - container images, 562–565*
 - docker images command, 562, 565*
 - docker ps command, 562–563*
 - docker run mypython command, 565*
 - docker search command, 563*
 - Dockerfiles, 564–565*
 - documentation, 565*
 - images, 564–565*
 - legacy rules, 566*
- Docker Swarm, 561
- DR, 50
- Duo Security, 167
- email security
 - AMP, 582*
 - CASE, 582*
 - encryption, 583*
 - ESA, 582*
 - FED, 583*
 - Office 365, 583–584*
 - SPF, 583*
 - Talos, 582*
 - WSA, 582*
- employee training, 49
- encryption, 49
- hybrid clouds, 48, 552
- IaaS, 48, 552
- IoT, 53
- issues/concerns, 48–50
- Kubernetes, 559
 - application deployments, 568*
 - clusters, 565–566, 568–570*
 - components of, 566*
 - deployments, 566–567*
 - displaying nodes, 567*
 - DNS servers, 570*
 - GKE, 568*
 - GUI, 570*
 - kubeadm, 567*
 - kubectrl get nodes command, 567, 570*
 - kubectrl version command, 567*
 - managing nodes, 568*
 - minikube start command, 566–567*
 - proxies, 570*
 - rules, 566*
 - starting, 566–567*
 - Stealthwatch Cloud and, 590*
 - tutorials, 568*
 - version verification, 567*

- long-term viability, 50
- man-in-the-middle attacks, 50
- micro-segmentation, 570–571
- microservices, 570–571
- Nomad, 560
- PaaS, 48, 552
- patch management, 575
- private clouds, 48, 552
- provider liability, 50
- public clouds, 48, 552
- regulatory requirements, 49
- SaaS, 48, 552
- security, 51
- security assessments, 575–577
- serverless cloud computing, 559
- session hijacking, 50
- session riding, 50
- side-channel attacks, 51
- SLA, 49
- SP 500–292, 48, 552
- SP 800–145, 47–48
- SQL injection, 50
- Stealthwatch Cloud, 251–256, 590
- Tetration, 593–594
 - ADM*, 594
 - connectors*, 595
 - Forensics feature*, 594
 - Security Dashboard*, 594–595
 - Security Score*, 595
 - Vulnerability Dashboard*, 595–596
- Umbrella, 167, 577
 - architecture of*, 577–578
 - Investigate*, 580–582
 - SIG*, 578–580
- waterfall development methodology, 552–553
- WebEx, 167
- XSS, 50
- clusters, 16, 425–427
- CNA (CVE Naming Authorities), 9–10
- CoA (Change of Authorization), 193–196
- coding, bugs in, 376
- collecting data, IoT, 51
- collision resistance, 85
- command injections, 32
- commercial CA, 94
- communication (covert), 23–24
 - application layer tunneling, 25
 - covert channels, 24
 - covert storage channel attacks, 23
 - covert timing channel attacks, 23
 - DNS, 25
 - HTTP, 25
 - HTTPS, 25
 - ICMP, 24
 - IPv6, 24
 - TCP, 24–25
 - UDP, 25
- community clouds, 48, 552
- confidentiality
 - CIA triad, 40–43
 - disclosure of confidential information, 12–13
- configurations
 - client-based remote access SSL VPN, 525
 - configuration files, security, 364–365
 - CoPP, 381–382
 - DAI, 331–332
 - DHCP snooping, 329–330
 - DMVPN
 - hub configurations*, 487–488
 - spoke configurations*, 488–489

- Flexible NetFlow, 275
 - flow exporters*, 280–282
 - flow monitors*, 278–282
 - flow records*, 276–278
- flow exporters, 280–282
- flow monitors, 278–282
- hub configurations, DMVPN, 487–488
- IPv4, flow monitors, 278–280
- IPv6, 367
 - autoconfiguration*, 376
 - flow monitors*, 278–280
 - routing*, 370–372
- misconfigurations, 9
- NetFlow, 269–270
- NTP, 363–364
- PortFast, 321–322
- RADIUS authentication, 202–205
- RSTP, 321–322
- site-to-site VPN
 - ASA firewall configurations*, 502–511
 - router configurations*, 479–502
- spoke configurations, DMVPN, 488–489
- SSL VPN, 516–518
- syslog, 362–363
- TACACS+ access, 196–199, 200–202
- WCCP
 - ASA configurations*, 609–610, 612
 - switch configurations*, 610–612
 - web traffic redirection to WSA*, 609–610, 612
- connectors**
 - AMP for Endpoints, 648
 - Tetration, 595
- console cables, management plane (NFP)**, 339
- containers**, 561
 - Apache Mesos, 560
 - Docker Swarm, 561
 - images, 561–565
 - Katacoda container deployments, 563
 - Kubernetes, 559
 - Nomad, 560
 - OCI, 561
 - orchestration, 559–561
 - registries, 561
- containment/eradication/recovery phase (IRP)**, 62
- content security**
 - AsyncOS, 604
 - ESA, 582, 619
 - Content SMA*, 624–628
 - deployments*, 620–621
 - DKIM*, 623
 - DLP*, 622–623
 - listeners*, 621–622
 - RAT*, 622
 - SenderBase*, 622
 - SMTP authentication/encryption*, 623
 - SPF*, 623
 - fundamentals of, 603–604
 - WSA, 582, 604
 - Content SMA*, 624–628
 - DNS*, 607
 - explicit forward mode*, 606–608
 - features of*, 604–605
 - policy configurations*, 615–617
 - policy-based routing*, 612–613
 - proxies*, 605–606
 - reports*, 617–619
 - security services*, 613–614
 - SOCKS proxies*, 607–608

- traffic redirection*, 609–610, 612–613
- transparent mode*, 608–609
- WCCP*, 608–612, 615
- web proxy IP spoofing*, 614–615
- WPAD*, 607
- content-dependent access control, 172
- Content SMA (Security Management Appliance), 624–628
- context-dependent access control, 172
- context services (ISE), 184–185
- continuity (business). *See* BCP
- Contiv, 120, 571
- contracts (cloud computing), ending, 50
- control information exchanges (TCP), 24
- control plane
 - NFP, 333–334
 - best practices*, 336–337
 - CoPP*, 336
 - CPPr*, 336–337
 - minimizing traffic*, 379–380
 - secure routing protocols*, 379
 - security*, 336–337
 - SPD*, 337
- traditional networks, 109
- controllers, SDN, 110
- cookies, manipulation attacks, 37–38
- coordination, incident response, 64, 72
- CoPP (Control Plane Policing), 336, 380–381
 - ACL, permitted traffic, 381
 - configurations, 381–382
 - show policy-map control-plane command, 380
 - verifying configurations, 382
- CORBA (Common Object Request Broker Architecture), 38
- covert communication, 23–24
 - application layer tunneling, 25
 - covert channels, 24
 - covert storage channel attacks, 23
 - covert timing channel attacks, 23
 - DNS, 25
 - HTTP, 25
 - HTTPS, 25
 - ICMP, 24
 - IPv6, 24
 - TCP, 24–25
 - UDP, 25
- CPPr (Control Plane Protection), 336–337, 383
- crackers, 13
- cracking passwords, 33–34
- credentials, authentication-based vulnerabilities
 - brute force attacks, 33–34
 - default credentials, 34
- credit card data, Trojans, 20
- crime (organized), 13
- CRL (Certificate Revocation Lists), 98, 100
- cross-certifying CA, 102
- Cross-Site Request Forgery (XSRF), 37
- Cross-Site Scripting (XSS), 32, 35–36
 - cloud computing, 50
 - DOM-based attacks, 36
 - examples of, 36
 - finding vulnerabilities, 36–37
 - reflected XSS attacks, 36
 - stored (persistent) XSS attacks, 36
 - testing, 37
- CRS (Composite Risk Scores), Cloudlock, 589
- crypters, malware distribution, 22
- crypto maps, 479, 506–508

CryptoDefense, 23**cryptography (cryptology)**

- algorithms, 470. *See also* ciphers
 - asymmetric algorithms, 83–84*
 - authentication-based vulnerabilities, 33*
 - symmetric encryption algorithms, 82–83*
- attacks, cloud computing, 50
- CA, 87–89, 93–94
 - authentication, 98–99*
 - commercial CA, 94*
 - digital certificates, 94, 97–98*
 - enrollment, 98–99*
 - identity certificates, 94, 96–97*
- ciphers. *See also* algorithms
 - block ciphers, 82*
 - ciphertext streams, 82*
 - defined, 80*
 - digit streams, 82*
 - polyalphabetic method, 81*
 - stream ciphers, 82*
 - substitution method, 81*
 - transposition method, 81*
- defined, 80
- DH key exchange protocol, 83–84
- digital certificates
 - in practice, 100*
 - revoking, 98–100*
- digital signatures, 86–89, 91–92, 93–94
- DSA, 84
- ECC, 84
- ElGamal asymmetric encryption, 84
- hashes, 84–86
- IPsec, 90
- keys, 81
 - digital certificates, 97*
 - keyspace, 89*

- managing, 89*

- OTP, 81–82*

- private key pairs, 93*

- public key pairs, 93*

- next-generation encryption protocols, 89–90

- PKCS, 83, 99

- PKI, 87, 93

- cross-certifying CA, 102*

- hierarchical CA, 101–102*

- single root CA, 101*

- subordinate CA, 101*

- topologies, 101–102*

- private key cryptography, 83, 93

- public key cryptography, 83

- digital certificates, 97*

- PKCS, 99*

- public key pairs, 93*

- quantum computing, 86

- RSA algorithm, 83

- SSL, 91

- TLS, 91

CryptoLocker, 23**CryptoWall, 23**

- CSIRT (Computer Security Incident Response Teams), 64–66, 71–72

CSP (Cloud Service Providers)

- customer/provider cloud security responsibilities, 573–575

- penetration testing, 575–577

- questions to ask, 575–577

CSRF. *See* XSRF

- CTA (Cognitive Threat Analytics), 262–268

- custody (digital forensics), chain of, 59

- custom privileges, 344

- customer/provider cloud security responsibilities, 573–575

CVE (Common Vulnerabilities and Exposures), 9–10, 30

CVSS (Common Vulnerability Scoring System), 67–71, 193, 595

cyberattacks, 12

cybersecurity, 6

access control management

DAC, 47

factors, 47

MAC, 47

privileges, 45–46

process of, 46

RBAC, 47

assets, defined, 12

CIA triad, 40

availability, 43–45

confidentiality, 40–43

integrity, 42–43

cloud computing, 50, 51

access, 49

API attacks, 51

attacks, 50–51

auditing, 49

authentication attacks, 51

BCP, 50

contracts, 50

cryptographic attacks, 50

data classification systems, 49

data separation, 49

DDoS attacks, 50

DR, 50

employee training, 49

encryption, 49

issues/concerns, 48–50

long-term viability, 50

man-in-the-middle attacks, 50

provider liability, 50

regulatory requirements, 49

session hijacking, 50

session riding, 50

side-channel attacks, 51

SLA, 49

SQL injection, 50

XSS, 50

covert communication, 23–25

CVE, 9–10

digital forensics, 58–59

exploits, 10–11

FIPS, 7

hardware vulnerabilities

authentication-based

vulnerabilities, 32–35

buffer overflows, 39

cookie manipulation attacks, 37–38

CVE, 30

injection vulnerabilities, 30–32

NVD, 30

OWASP Top 10 list, 40

race conditions, 38

ret2libc attacks, 39

unprotected API, 38–39

XSRF, 37

XSS, 35–37

incident response, 55

benefits of, 56

CERT, 71–72

coordination centers, 72

CSIRT, 64–66, 71–72

CVSS, 67–71

digital forensics, 58–59

DIH, 73

false positives/negatives, 57–58

FIRST, 71

- FISMA of 2002, Public Law 107-347*, 56
- incidents, defined*, 56–57
- incidents, examples of*, 57
- incidents, reporting*, 58–59
- incidents, security levels*, 58
- information sharing/coordination*, 64
- IRC, 73
- IRP, 60–63
- IRT, 73–74
- ISO/IEC 27002:2013, 55–56
- MSSP, 73
- NIST, 55–56
- PSIRT, 66–67, 70
- SDL, 70–71
- SP 800–61, 56, 61
- SP 800–61 revision 2*, 55, 60
- SP 800–83, 55
- SP 800–86, 55
- tabletop exercises/playbooks*, 63–64
- TPS security*, 71
- true positives/negatives*, 57–58
- InfoSec vs., 7
- IoT, security challenges/considerations, 52
- IRP, 29, 60–61, 63
- ISO/IEC 27000 series, 8
- ITL, 8
- keyloggers, 25–26
- malware
 - distribution types*, 22
 - dynamic analysis*, 27–29
 - payloads*, 17–18
 - static analysis*, 27–29
 - transmission methods*, 16–17
- NIST cybersecurity framework, 7–8
- NISTIR, 8
- open source software vulnerabilities, 40
- ransomware (data hiding), 19, 23
- risk
 - defined*, 12
 - residual risk*, 12
- software vulnerabilities
 - authentication-based vulnerabilities*, 32–35
 - buffer overflows*, 39
 - cookie manipulation attacks*, 37–38
 - CVE, 30
 - injection vulnerabilities*, 30–32
 - NVD, 30
 - OWASP Top 10 list, 40
 - race conditions*, 38
 - ret2libc attacks*, 39
 - unprotected API*, 38–39
 - XSRF, 37
 - XSS, 35–37
- SP
 - 800 Series*, 7
 - 1800 Series*, 8
- spyware, 16, 26–27
- threats
 - defined*, 9, 12–13
 - threat actors*, 13–14
 - threat intelligence*, 14–15
- Trojans
 - communication methods*, 19
 - defined*, 18
 - effects of*, 22
 - goals of*, 20
 - infection mechanisms*, 20–21
 - ports*, 19
 - types of*, 18–19

- viruses, 16
 - components of*, 17–18
 - transmission methods*, 16–17
 - types of*, 16–17
- vulnerabilities, defined, 9–10
- worms, 16
 - transmission methods*, 16–17
 - types of*, 16–17
- zero-trust, 161–167

CyBOX (Cyber Observable EXpression), 15

D

- DAC (Discretionary Access Controls), 47, 168
- dACL (downloadable ACL), 181
- DAI (Dynamic ARP Inspection), 324, 330–332, 338
- “dark web”, 10
- DAST (Dynamic Application Security Testing), 572–573
- database view, 172
- data breaches
 - examples of, 156
 - IOC, 454, 455
- data center, NetFlow deployment scenario, 246–248
- data classification systems, cloud computing, 49
- data collection, IoT, 51
- data-driven network segmentation, 286–288
- data hiding (ransomware), 19, 23
- data integrity, verifying, 84–86
- data leak detection/prevention, NetFlow, 231
- data plane
 - NFP, 333–334
 - best practices*, 337–338
 - IPv6 configuration/security*
 - security*, 337–338
 - traditional networks, 109
- data separation, cloud computing, 49
- data storage, Trojans, 20
- DCE/RPC preprocessors, 450
- DCOM (Distributed Component Object Model), 38
- DDoS attacks, 13
 - availability (CIA triad), 45
 - botnets, 45
 - cloud computing, 50
 - NetFlow and DDoS attack migration, 229–231
- debug commands
 - AAA for Cisco routers, troubleshooting, 353–356
 - IPsec tunnels, troubleshooting, 496–502
 - site-to-site VPN router configurations, 496–502
- debugging TACACS+, 199–200
- default credentials, authentication-based vulnerabilities, 34
- deployment scenarios, NetFlow, 242–243
 - data center, 246–248
 - Internet edge, 245
 - remote VPN, 248–249
 - site-to-site VPN, 248–249
 - user access layer, 243
 - WLAN, 244
- detection and analysis phase (IRP), 61–62
- development methodologies (cloud computing)
 - agile methodology, 553–556, 559
 - waterfall methodology, 552–553

- device hardening
 - IPv4, 372
 - IPv6, 372
- device tracking, IPv6, 377
- DevNet, 136, 142
- DevOps, 552, 556–557
- DevSecOps, 571
 - assurance tools/methods, 572–573
 - CI/CD pipelines, 572
 - OWASP Proactive Controls, 571–572
 - tutorials, 571
- DH key exchange protocol, 83–84
- DHCP (Dynamic Host Configuration Protocol)
 - ASA and, 412
 - DHCPv6, 375
 - snooping, 324, 328–330, 338
- DHS (Department of Homeland Security), CERT, 72
- Diameter, 176–178, 179
- Diffie-Hellman key exchange, 471–473, 504, 507, 509
- digit streams (ciphers), 82
- digital certificates, 94
 - AAA, 100
 - components of, 97–98
 - CRL, 98, 100
 - identity certificates, 94, 96–97
 - OCSP, 100
 - in practice, 100
 - revoking, 98–100
 - root certificates, 95–96
 - thumbprint algorithms, 98
- digital/electronic wallets, Trojans, 20
- digital forensics, 58–59
- digital signatures, 86–89, 91–92
 - digital certificates, 97
 - DSA, 84
 - RSA, 93–94
- DIH (Designated Incident Handlers), 73
- DIKTA questions, exam preparation, 658
- direct DoS attacks, availability (CIA triad), 44
- direct objects
 - insecure direct object reference vulnerabilities, 35
 - reference example, 35
- directories, X.500 standards, 97
- disaster recovery (DR), cloud computing, 50
- disclosure of confidential information, 12–13
- distributed ISE deployments, sizing, 214
- DKIM (Domain Keys Identified Mail), 623
- DLP (Data Loss Prevention), ESA, 622–623
- DMVPN (Dynamic Multipoint Virtual Private Networks), 486
 - example of, 487
 - hub configurations, 487–488
 - NAT-T, 487
 - NHRP, 486–487
 - site-to-site VPN router configurations, 486–489
 - spoke configurations, 488–489
- DNA (Digital Network Architecture)
 - application policies, 126–127
 - architecture of, 121
 - DNAC, 121–124
 - API, 130, 132
 - Assurance solution, 128–129
 - multivendor support, 132
 - Security solution, 132
 - group-based access control policy, 124
 - ip-based access control policy, 124

- policies, 123
- traffic copy policy, 127
- DNS (Domain Name System)**
 - attacks, cloud computing, 50
 - covert communication, 25
 - DNS preprocessors, 450
 - MX records, 620
 - OpenDNS, Umbrella, 577–582
 - servers, Kubernetes, 570
 - Umbrella, 577
 - architecture of*, 577–578
 - Investigate*, 580–582
 - SIG*, 578–580
 - WSA, 607
- dnscat**, 25
- do not allow negotiations, VLAN**, 323
- Docker**
 - container images, 562–565
 - docker images command, 562, 565
 - docker ps command, 562–563
 - docker run mypython command, 565
 - docker search command, 563
 - Dockerfiles, 564–565
 - documentation, 565
 - images, 564–565
 - legacy rules, 566
- Docker Swarm**, 561
- documentation**
 - API, 39
 - Docker, 565
 - FIPS, 7
 - ISO/IEC 27000 series, 8
 - ITL bulletins, 8
 - NISTIR, 8
 - SP 800 Series, 7, 8
 - Swagger, 39
 - WADL, 39
 - WSDL, 39
 - XSD documents, 38
- DOM (Document Object Model)**
 - cookie manipulation attacks, 37–38
 - XSS attacks, 36
- DoS (Denial of Service) attacks**, 13
 - amplification attacks, availability (CIA triad), 45
 - availability (CIA triad), 44–45
 - buffer overflows, availability (CIA triad), 45
 - data plane (NFP), 338
 - DDoS attacks, availability (CIA triad), 45
 - direct DoS attacks, availability (CIA triad), 44
 - IPv4, 373–374
 - IPv6, 373–374
 - reflected DoS attacks, availability (CIA triad), 45
 - Trojans, 19
- DR (Disaster Recovery), cloud computing**, 50
- droppers**
 - malware distribution, 22
 - spyware, 26
- DSA (Digital Signature Algorithm)**, 84
- DTLS (Datagram Transport Layer Security)**, 529
- dual stacks, IPv6**, 376
- Duo Access Gateways**, 160
- Duo Security**, 159–163
 - cloud computing, 167
 - SSO applications, 166
- duties, separation of**, 155
- dynamic malware analysis**, 27–29
 - FakeNet, 29
 - MAC addresses, 29
 - VM, 28–29

dynamic NAT and ASA, 438, 441–442
dynamic PAT and ASA, 442

E

EAP (Extensible Authentication Protocol), 179

EAPoL (EAP over LAN), 179

eavesdropping attacks, IPv4/IPv6, 374

e-banking, 19

ECC (Elliptic Curve Cryptography), 84

edb (Evan's Debugger), 27

EDR (Endpoint Detection and Response), 638

EER (Equal Error Rates), 159

EIGRP, MD5 authentication, 384–385

electronic/digital wallets, Trojans, 20

ElGamal asymmetric encryption, 84

email security

AMP, 582

attachments, Trojans, 21

CASE, 582

DNS MX records, 620

encryption, 583

ESA, 582, 619

FED, 583

IMAP, 620

MDA, 619

MSA, 619

MTA, 619

MUA, 619

Office 365, 583–584

POP, 620

SPF, 583

Talos, 582

WSA, 582

employee training, cloud computing, 49

encrypted management protocols, management plane (NFP), 340

encryption, 86

cloud computing, 49

components of, 92

email security, 583

encrypted management protocols, 344–345

IDS, 58

IPS, 58

next-generation encryption protocols, 89–90

SMTP, ESA, 623

symmetric encryption algorithms, 82–83

endpoint protection/detection, 636–637

AMP for Endpoints, 637–639

AnyConnect AMP Enabler, 650

Application Control, 644–645

connectors, 648

Ethos, 650

exclusion sets, 645–647

IP blacklists/whitelists, 643–644

Outbreak Control, 639–643

policies, 648–649

reports, 651–654

Spero, 650

TETRA, 650

EDR, 638

EPP, 638

ETDR, 637

Threat Response, 654–655

enforcers, networks as, 226–227

enrollment, CA, 98–99

EPG (Endpoint Groups), 289–290

EPP (Endpoint Protection Platform), 638

errors

CER, 159

EER, 159

FAR, 159

FRR, 159

**ERSPAN mode (passive), NGFW/
NGIPS, 422****ESA (Email Security Appliance), 582,
619**

Content SMA, 624–628

deployments, 620–621

DKIM, 623

DLP, 622–623

listeners, 621–622

RAT, 622

SenderBase, 622

SMTP authentication/encryption, 623

SPF, 623

**ETA (Encrypted Traffic Analytics),
132, 262****ETDR (Endpoint Threat Detection and
Response), 637****EtherType ACL, 431****ethical hackers, 13****Ethos, 454, 650****Evan's Debugger (edb), 27****exams**

preparing for, 658

*Blueprints, 658**DIKTA questions, 658**hands-on activities, 658**Pearson Cert Practice Test
engine, 659**“Review Questions” sections,
659**review/study plans, 658–659*

updates, 686–687

**exchanging control information
(TCP), 24****exclusion sets, 645–647****explicit forward mode (WSA),
606–608****exploits**

“dark web”, 10

defined, 10–11

Exploit-DB, 10

GitHub, 10

POC exploits, 10

searchsploits, 10–11

zero-day exploits, 10

exposures, CVE, 9–10**extended ACL, 430–431****F**

**factors (access control management),
47****failover (high-availability) mode,
ASA/FTD, 423–425****FakeNet, 29****false positives/negatives, incident
response, 57–58****FAR (False Acceptance Errors), 159****fast infections, 17****FDM (Firepower Device Manager),
404–407****FED (Forged Email Detection), 583****federated identities, 165–166****FFRDC (Federally Funded Research
and Development Center), 9–10****file infections, 16****filtering**

bogus IPv6 addresses, 376

ICMP, 437

ICMPv6, 377

nonlocal multicast addresses, 377

- traffic
 - ASA, 396–397
 - to-the-box traffic filtering*, 434–435
 - through-the-box traffic filtering*, 431
- Findsecbugs, 572–573
- FIPS (Federal Information Processing Standards), 7
- Firepower, 396, 398
 - 1000 series, 397
 - 2100 series, 397–398
 - 4100 series, 398
 - 9300 series, 399
 - FDM, 404–407
 - FMC, 401–404
 - FXOS, 407
 - NGIPS variables, 449–450
 - platform settings policies, 450
 - software patches/updates, 458
- FirePOWER module, 396–397
- firewalls, 180, 395–396
 - ASA firewalls, 187
 - CDO, 408–410
 - Centri Firewall, 396
 - Cisco history/legacy, 396
 - Firepower, 398
 - 1000 series*, 397
 - 2100 series*, 397–398
 - 4100 series*, 398
 - 9300 series*, 399
 - MMTF, 416
 - NGFW
 - inline pairs*, 420
 - inline pairs with tap*, 420–421
 - passive ERSPAN mode*, 422
 - passive (monitoring) mode*, 420–422
 - partitioning, 414
 - routed firewalls, 413
 - security contexts, 414
 - SMTF, 414–416
 - transparent firewalls, 413, 414
 - MMTF*, 416
 - SMTF*, 414–416
 - ZBFW, 411–412
- FIRST (Forum of Incident Response and Security Teams), 71
- first-hop security binding tables, 377
- FISMA (Federal Information Security Management Act) of 2002, Public Law 107–347, 56
- five-tuples, 227
- Flame, 17
- Flex-Auth (Flexible Authentication), 203
- Flexible NetFlow, 228
 - application tracking (simultaneous), 270–271
 - configurations, 275
 - flow monitors*, 278–282
 - flow records*, 276–278
 - flow exporters, 275, 280–282
 - flow monitors, 275, 282–283
 - flow samplers, 275
 - IPFIX export format, 283
 - key fields, 271–273
 - non-key fields, 273–274
 - records, 271
 - flow records*, 276–278
 - predefined records*, 274
 - user-defined records*, 275
- FlexVPN, 492–496, 499–501
- flow
 - defined, 227

- Flexible NetFlow
 - flow exporters*, 275, 280–283
 - flow monitors*, 275
 - flow samplers*, 275
- FlowCollector, 250
- flow exporters
 - configurations*, 280–282
 - NX-OS configurations*, 284
 - show flow exporter command*, 281
 - show running-config flow exporter command*, 281
- flow monitors
 - applications*, 285
 - applying to interfaces*, 282–283
 - configurations*, 278–282
 - NX-OS configurations*, 284
 - show flow monitor command*, 279
 - show flow monitor name*
 - NY-ASR-FLOW-MON-1*
 - cache record format command*, 281–282
 - show running-config flow monitor command*, 279–280
- flow records
 - configurations*, 276–278
 - NX-OS configurations*, 284
- FlowReplicator, 251
- FlowSensor, 251
- fps, determining, 269
- inline pairs, 420
- IPFIX, 237–241, 283
- licenses, 250
- NetFlow, 225–237
- sessions versus, 229
- Flow Sensor (Stealthwatch)**, 233
- FMC (Firepower Management Center)**, 401–404, 449
- fog computing, 51
- fog-edge devices, IoT, 52
- forensics
 - digital forensics, 58–59
 - Forensics feature (Tetration), 594
 - network security, NetFlow, 231–236
- four-step shutdowns (TCP)**, 25
- fps (Flow Per Second), determining, 269
- fragmentation
 - IDS, 58
 - IPS, 58
 - IPv6, 380
 - site-to-site VPN configurations, 510–511
- freeware, Trojans, 21
- FRR (False Rejection Errors)**, 159
- FTD (Firepower Threat Defense)**, 397
 - access control policies, 443–445
 - BVI and, 417–419
 - CDO, 409–410
 - clustering, 425–427
 - deployment design considerations, 422–423
 - deployment modes, 416–417
 - failover (high-availability) mode, 423–425
 - FDM, 404–407
 - Firepower, 396, 398
 - 1000 series*, 397
 - 2100 series*, 397–398
 - 4100 series*, 398
 - 9300 series*, 399
 - FMC, 401–404
 - FXOS, 407
 - inline interfaces, 420
 - interface modes, 417–419
 - intrusion policies, 446–449
 - ISR and, 399
 - remote-access VPN, 530–531, 540
 - site-to-site VPN, 541–543

FTP (File Transfer Protocol)
 Telnet preprocessors, 450
 Trojans, 19
fuzz testing (fuzzing), 573
FXOS (Firepower eXtensible Operating System), 407

G

GDOI protocol, 489
geolocation updates, 458
GETVPN (Group Encrypted Transport VPN), 489–492
Ghidra, 28
GitHub, 10, 15
 agile development methodology (cloud computing), 555
 fuzz testing (fuzzing), 573
 GETVPN, 492
 IPsec VPN, 499
 pxGrid examples, 184
 XSS, 36
 ZBFW, 412
GKE (Google Kubernetes Engine), 568
global ACL, 429
global addresses, internal address translation to, 438
gNMI (gRPC Network Management Interface), 145–146
government/state-sponsored threats, 13
Grandcrab, 23
GraphQL, 38, 137
gray hat hackers, 14
GRE, site-to-site VPN router configurations
 GRE over IPsec, 482–484
 mGRE tunnels, 486

group-based access control policy (DNA), 124
group policy
 client-based remote access SSL VPN, 525–526
 remote-access VPN ASA configurations, 513
 SSL VPN, 518–519
GTP preprocessors, 451
guest access (unauthenticated/ authenticated), 188

H

hackers
 attacks, 12
 black hat hackers, 14
 defined, 13
 ethical hackers, 13
 gray hat hackers, 14
 motivations, 14
hacking, IoT hacking tools/methods, 54–55
hacktivists, 13
handshakes (three-step), TCP, 24
hardening devices, IPv4/IPv6, 372
hardware vulnerabilities, 9
 authentication-based vulnerabilities, 32–35
 buffer overflows, 39
 cookie manipulation attacks, 37–38
 CVE, 30
 injection vulnerabilities, 30–32
 NVD, 30
 OWASP Top 10 list, 40
 race conditions, 38
 ret2libc attacks, 39
 unprotected API, 38–39

- XSRF, 37
- XSS, 35–37
- hashes, 86
 - AMP, 85
 - collision resistance, 85
 - cryptographic hash functions, 85
 - defined, 84
 - example of, 84–85
 - HMAC, 86
 - MD5
 - checksums*, 85
 - md5sum Linux command*, 85
 - SHA
 - checksums*, 85
 - SHA512 checksum*, 84
 - sbasum Linux command*, 85
 - vulnerabilities*, 85–86
 - verify md5 command, 84
- hierarchical CA, 101–102
- high-availability (failover) mode,
 - ASA/FTD, 423–425
- hijacking sessions, 34, 50
- HMAC (Hashed Message Authentication Code), 86
- hoaxes (virus), 17
- hop-by-hop extension headers, IPv6, 375–376
- HTML injection vulnerabilities, 32
- HTTP (HyperText Transfer Protocol)
 - ACL and HTTP traffic, 432–433
 - covert communication, 25
 - HTTP preprocessors, 450
 - Requests, 135
 - status code messages, 135
 - XSRF, 37
- HTTPS (HTTP Secure), 25, 91–92, 362
- hub configurations, DMVPN, 487–488
- hybrid clouds, 48, 552
- IaaS (Infrastructure as a Service), 48, 552
- ICMP (Internet Control Message Protocol)
 - covert communication, 24
 - filtering, 437
 - icmp command, 437
 - ICMPv6
 - filtering*, 377
 - IPv6*, 376
 - unreachables, 380
- IDA Pro, 27
- identity certificates, 94, 96–97
- identity management
 - 802.1X, 187
 - active policy enforcement*, 295–298
 - authentication configurations*, 205–211
 - authentication failures*, 203
 - C3PL*, 204–205
 - monitor mode deployments*, 294–295
 - Multi-Auth mode*, 203
 - Open Authentication*, 204
 - port security*, 203
 - CoA, 193–196
 - Flex-Auth, 203
 - ISE, 181–182
 - authorization rules*, 187–188
 - context services*, 184–185
 - design tips*, 211–213
 - identity services*, 184–185, 187–188
 - profiling services*, 184–187
- MAB, 188
- posture assessments, 192–193

- pxGrid, 182–184
- RADIUS, 187, 202–205
- TACACS+
 - access configuration, 196–199, 200–202*
 - debugging, 199–200*
- TrustSec, 190–192
 - ACI integration, 298–301*
 - active policy enforcement, 295–298*
 - monitor mode deployments, 294–295*
 - SGT, 188*
- web authentication, 187–188
- identity NAT and ASA, 442**
- identity services (ISE), 184–185, 187–188**
- IDS (Intrusion Detection Systems), 399**
 - address proxying, 58
 - address spoofing, 58
 - encryption, 58
 - false positives/negatives, 57–58
 - fragmentation, 58
 - low-bandwidth attacks, 58
 - pattern change evasion, 58
 - true positives/negatives, 57–58
- IEC. *See* ISO**
- IKE (Internet Key Exchange), 470**
 - IKEv1 phase 1, 470–472
 - IKEv1 phase 2, 472–474
 - IKEv2, 475–476, 504
 - NAT-T, 474
 - RFC 2409, 470
 - RFC 5996, 470
 - UDP, 472
- IM (Instant Messaging), Trojans, 20**
- IMAP (Internet Message Access Protocol), 451, 620**
- impersonated mobile apps, Trojans, 21**
- implicit deny, authorization, 168**
- incident response, 55**
 - benefits of, 56
 - CERT, 71–72
 - coordination centers, 72
 - CSIRT, 64–66, 71–72
 - CVSS, 67–71
 - digital forensics, 58–59
 - DIH, 73
 - false positives/negatives, 57–58
 - FIRST, 71
 - FISMA of 2002, Public Law 107–347, 56
 - incidents
 - defined, 56–57*
 - examples of, 57*
 - reporting, 58–59*
 - security levels, 58*
 - information sharing/coordination, 64
 - IRC, 73
 - IRP
 - containment/eradication/recovery phase, 62*
 - defined, 60–61*
 - detection and analysis phase, 61–62*
 - elements of, 60*
 - phases of, 61–63*
 - post-incident activity phase, 63*
 - preparation phase, 61*
 - process of, 61–63*
 - IRT, 73–74
 - ISO/IEC 27002:2013, 55–56
 - MSSP, 73
 - NetFlow, 231–236
 - NIST, 55–56
 - PSIRT, 66–67, 70

- SDL, 70–71
- SP 800–61, 56, 61
- SP 800–61 revision 2, 55, 60
- SP 800–83, 55
- SP 800–86, 55
- tabletop exercises/playbooks, 63–64
- TPS security, 71
- true positives/negatives, 57–58
- infection routines, 18**
- inferential (blind) SQL injection, 32**
- information sharing/coordination, incident response, 64**
- InfoSec (Information Security) vs cybersecurity, 7**
- infrastructure access controls, 170**
- infrastructure security**
 - AAA, 342
 - CLI, troubleshooting AAA for Cisco routers, 353–356*
 - method list, 343*
 - router access authentication, 342–343*
 - user authentication, 349–353*
 - administrator views, 344
 - bootsets, 364–365
 - Cisco IOS, 346–347, 364–365
 - Cisco IOS-XE, 346–347
 - Cisco IOS-XR, 346–347
 - Cisco NX-OS, 346–347
 - configuration files (startup), 364–365
 - control plane
 - CoPP, 380–382*
 - CPPr, 383*
 - minimizing traffic, 379–380*
 - packets, 379*
 - encrypted management protocols, 344–345
 - HTTPS, 362
 - IPv4
 - best practices, 372–373*
 - common threats, 373–374*
 - IPv6, 365–366, 374–375
 - ACL, 377–378
 - address format, 367*
 - address types, 367–370*
 - best practices, 372–373, 376–377*
 - common threats, 373–374*
 - configurations, 367*
 - IPv4 versus, 366*
 - moving to, 372*
 - potential risks, 375–376*
 - router configurations, 370–372*
 - security plans, 372*
 - shortcuts, 367*
 - Layer 2 security, 310
 - BPDU Guard, 324–325*
 - CDP, 327–328*
 - common threats, 322–323*
 - DAI, 330–332*
 - DHCP snooping, 328–330*
 - LLDP, 327–328*
 - port security, 325–327*
 - Root Guard, 325*
 - security toolkit, 324*
 - STP, 317–322*
 - VLAN, 310–317*
 - logging features, 362–363
 - logging files, 345–346
 - NFP, 332
 - control plane, 333–334, 336–337*
 - data plane, 333–334, 337–338*
 - framework of, 333*
 - importance of, 332*
 - interdependence, 333*

- management plane*, 333–336, 338–341
- passwords*, 338–341
- NTP, 346
 - authentication keys*, NTPv3, 363–364
 - client synchronization*, 364
 - configurations*, 363–364
- passwords, 341, 347–348
- privileges (custom), 344
- RBAC, 344
 - parser views*, 358–360
 - privilege levels*, 356–358
- routing protocols, 383
 - BGP*, 386–387
 - EIGRP*, 384–385
 - MD5 authentication*, BGP, 386–387
 - MD5 authentication*, RIPv2, 385–386
 - OSPF*, 383–384
 - RIP*, 385–386
- SSH, 360–362
- startup configuration files, 364–365
- Syslog, 362–363
- user authentication, 349–353
- injection vulnerabilities**, 30
 - command injections, 32
 - HTML, 32
 - SQLi, 30, 31–32
 - in-band SQL injection*, 32
 - blind (inferential) SQL injection*, 32
 - example of*, 31
 - out-of-band SQL injection*, 32
 - queries*, 32
 - SQL statements*, 30–31
- inline interfaces**, 420
 - inline pairs**, 420
 - flow, 420
 - with tap, 420–421
 - insecure direct object reference vulnerabilities**, 35
 - insider information**, Trojans, 20
 - INSTEON**, IoT, 54
 - integrity**
 - CIA triad, 42–43
 - data, verifying, 84–86
 - Intent API**, DNAC API, 130, 132
 - interface ACL**, 429
 - internal to global address translation**, 438
 - Internet edge**, NetFlow deployment scenario, 245
 - Internet of Things**. *See* IoT
 - inter-VLAN routing**, 316
 - router-on-a-stick, 316–317
 - virtual “sub” interfaces, 316–317
 - intrusions**
 - detection, defined, 446
 - policies (FTD), 446–449
 - prevention, defined, 446
 - IOC (Indicators of Compromise)**, 15, 454–455
 - IOS (Internetworking Operating System)**, 346–347
 - crypto maps, 479
 - NetFlow configurations, 269–270
 - VPN, site-to-site VPN configurations, 479–482
 - ZBFW, 411–412
 - IOS-XE**, 346–347
 - crypto maps, 479
 - NetFlow configurations, 269–270
 - site-to-site VPN configurations, 479–482

IOS-XR, 346–347**IoT (Internet of Things), 51**

- 6LoWPAN, 54
- API, 53
- BLE, 53
- Bluetooth Smart, 53
- cellular connections, 54
- closed-loop functioning, 51
- cloud computing, 53
- data collection, 51
- fog computing, 51
- fog-edge devices, 52
- hacking tools/methods, 54–55
- INSTEON, 54
- LoRaWAN, 54
- LRWPAN, 54
- messaging protocols, 54
- network resource preservation, 51
- protocols, 53–54
- security challenges/considerations, 52
- Wi-Fi, 54
- Zigbee, 53
- Z-Wave, 53

IP (Internet Protocol)

- accounting versus NetFlow, 229
- addresses, management plane (NFP), 335
- blacklists/whitelists, 643–644
- IP Source Guard, 324, 338
- pools, remote-access VPN ASA configurations, 513
- spoofing, web proxy IP spoofing (WSA), 614–615

ip-based access control policy (DNA), 126

- ip ospf authentication-key command, 383
- ip ospf message-digest-key command, 383

IPFIX (IP Flow Information Export), 237–238

- architecture of, 238
- Flexible NetFlow and IPFIX export format, 283
- mediators, 239
- SCTP, 241
- templates, 238
 - example of, 240*
 - option templates, 241*
 - structure of, 239–240*

IPS (Intrusion Prevention Systems), 395–396

- address proxying, 58
- address spoofing, 58
- encryption, 58
- false positives/negatives, 57–58
- fragmentation, 58
- legacy IPS, 399–400
- low-bandwidth attacks, 58
- NGIPS, 399–401
 - FMC, 401–404*
 - inline pairs, 420*
 - inline pairs with tap, 420–421*
 - passive (monitoring) mode, 420–422*
 - passive ERSPAN mode, 422*
 - preprocessors, 450–452*
 - variables, 449–450*
- pattern change evasion, 58
- true positives/negatives, 57–58

IPsec (IP security), 90

- GRE over IPsec, site-to-site VPN router configurations, 482–484
- IKE, 470
 - IKEv1 phase 1, 470–472*
 - IKEv1 phase 2, 472–474*
 - IKEv2, 475–476*

- NAT-T, 474
- RFC 2409, 470
- RFC 5996, 470
- UDP, 472
- OSPF over IPsec, 509
- remote-access VPN ASA
 - configurations, 512–514
- site-to-site VPN ASA firewall
 - configurations, 505–506
- transform sets, 479
- tunnels, troubleshooting, 496–502
- IPv4 (Internet Protocol version 4)**
 - AAA, 373
 - application layer attacks, 373–374
 - best practices, 372–373
 - common threats, 373–374
 - device hardening, 372
 - DoS attacks, 373, 374
 - eavesdropping attacks, 374
 - flow monitor configurations, 278–280
 - IPv6 versus, 366
 - man-in-the-middle attacks, 374
 - physical security, 372
 - routing attacks, 374
 - routing protocol security, 373
 - security policies, 373
 - sniffing attacks, 374
 - spoofing attacks, 374
 - unauthorized access, 374
 - zone access, 373
- IPv6 (Internet Protocol version 6), 365–366**
 - 6LoWPAN, IoT, 54
 - AAA, 373
 - ACL, 377–378
 - address format, 367
 - address shortcuts, 367
 - anycast addresses, 368–369
 - application layer attacks, 373–374
 - ARP requests, 380
 - autoconfiguration, 376
 - best practices, 372–373, 376–377
 - bugs in code, 376
 - common threats, 373–374
 - configurations, 367
 - covert communication, 24
 - data plane (NFP)
 - device hardening, 372
 - device tracking, 377
 - DHCPv6, 375
 - DoS attacks, 373, 374
 - dual stacks, 376
 - eavesdropping attacks, 374
 - extra addresses, 367–368
 - filtering
 - bogus addresses*, 376
 - ICMPv6*, 377
 - nonlocal multicast addresses*, 377
 - first-hop security binding tables, 377
 - flow monitor configurations, 278–280
 - fragmentation, 380
 - hop-by-hop extension headers, 375–376
 - ICMP unreachable, 380
 - ICMPv6, 376
 - interface information, 369–370
 - IPv4 versus, 366
 - link-local addresses, 368
 - loopback addresses, 368
 - man-in-the-middle attacks, 374
 - moving to, 365, 372
 - multicast addresses, 369
 - all-nodes multicast addresses*, 368

- all-routers multicast addresses*, 368
- solicited-node multicast addresses*, 369
- ND Inspection, 377
- NDP, 375
- neighbor cache resource starvation, 375
- packet amplification attacks, 376
- physical security, 372
- port-based access lists, 377
- potential risks, 375–376
- RA Guard, 377
- RH0 packets, 377
- rogue IPv6 devices, 377
- routing, 370–372
- routing attacks, 374
- routing protocol security, 373
- security, 374–375
- security plans, 372
- security policies, 373
- SeND, 377
- show ipv6 route command, 372
- sniffing attacks, 374
- spoofing attacks, 374
- TTL, 380
- tunneling, 376, 377
- unauthorized access, 374
- unicast addresses, 368–369
- zone access, 373
- IRC (Incident Response Coordinators)**, 73
- IRC (Internet Relay Chats), Trojans**, 21
- IRP (Incident Response Plans)**, 29
 - containment, eradication, recovery phase, 62
 - defined, 60–61
 - detection and analysis phase, 61–62
 - elements of, 60
 - phases of, 61–63
 - post-incident activity phase, 63
 - preparation phase, 61
 - process of, 61–63
- IRT (Incident Response Teams)**, 73–74
- ISAKMP, site-to-site VPN ASA firewall configurations**, 503–504
- ISE (Identity Services Engine)**, 181–182
 - 802.1X
 - active policy enforcement*, 295–298
 - monitor mode deployments*, 294–295
 - authorization rules, 188–190
 - context services, 184–185
 - design tips, 211–213
 - distributed deployments, sizing, 214
 - DNAC, 121–122, 124
 - identity services, 184–185, 187–188
 - network segmentation, 290–291
 - profiling services, 184–187
 - TACACS+ access configurations, 200–202
- ISO (International Organization for Standardization)**, 8
 - ISO/IEC 27000 series, 8
 - ISO/IEC 27001:2005, 66
 - ISO/IEC 27002:2005, 66
 - ISO/IEC 27002:2013, 55–56
 - ISO/IEC 27005:2008, 66
 - ISO/IEC 27033, 66
 - ISO/PAS 22399:2007, 66
- ISR (Integrated Service Routers), FTD for ISR**, 399
- issuers**
 - digital certificates, 97
 - root certificates, 95
- ITL bulletins**, 8

J - K

Kanban scheduling system, 555
 Katacoda container deployments, 563
 keychain authentication (BGP), 387
 KeyGhost, 26
 keyloggers, 25–26
 keys (cryptography), 81

- digital certificates, 97
- keyspace, 89
- managing, 89
- OTP, 81–82
- private key pairs, 93
- public key pairs, 93

 know (authorization), need to, 168
 knowledge, authentication by, 156–157
 Kubernetes (k8s), 559

- application deployments, 568
- clusters, 565–566, 568–570,
- components of, 566
- deployments, 566–567
- DNS servers, 570
- GKE, 568
- GUI, 570
- kubeadm, 568
- kubectl get nodes command, 567, 570
- kubectl version command, 567
- minikube start command, 566–567
- nodes
 - displaying*, 567
 - managing*, 568
- proxies, 570
- rules, 566
- starting, 566–567
- Stealthwatch Cloud and, 590

- tutorials, 568
- version verification, 567

KVM (Kernel-based Virtual Machines) and ISE, 182

L

labels (security), authorization, 167
 LAN (Local Area Networks)

- EAPoL, 179
- VXLAN, 110, 112–114
- WLAN, NetFlow deployment scenario, 244

 Layer 2 security, 310

- 802.1X, 324
- ACL, 324
- BPDU Guard, 324–325
- CDP, 327–328
- common threats, 322–323
- DAI, 324, 330–332
- DHCP snooping, 324, 328–330
- IP Source Guard, 324
- LLDP, 327–328
- loops, 317–318
- port security, 324, 325–327
- Root Guard, 324, 325
- Storm Control, 324
- STP, 317–318
 - annotations*, 318–320
 - instances of*, 321
 - new ports*, 321
 - port states*, 321
 - PortFast*, 321–322
 - RSTP*, 321–322
 - time until forwarding*, 321–322
 - verification*, 318–320
- toolkit, 324

- VLAN, 310–311
 - creating*, 311
 - defined*, 311
 - example of*, 311
 - show interfaces Gi0/2 switchport command*, 313
 - show vlan brief command*, 312
 - show vlan id command*, 312–313
 - switch ports*, 323
 - trunking*, 313–315
 - VLAN 10 interface assignments, 312
 - VLAN 20 interface assignments, 312
 - leaf switches
 - ACI, 110
 - spine nodes/switches, 110–111
 - leaks (data), detection/prevention with NetFlow, 231
 - least privilege, principle of, 155
 - liability (provider), cloud computing, 50
 - licenses (flow), 250
 - linked (centralized) identities, 165–166
 - link-local addresses, IPv6, 368
 - Linux
 - cat Linux command, 84
 - Duo Security, 161–162
 - md5sum Linux command, 85
 - shasum Linux command, 85
 - listeners (ESA), 621–622
 - LLDP (Link Layer Discovery Protocol), 327–328
 - locking down switch ports, 323
 - Login Password Retry Lockout, management plane (NFP), 339–340
 - logging
 - ACL, 380
 - files, 345–346
 - management plane (NFP), 340
 - NSEL, 248
 - syslog, configurations, 362–363
 - long-term viability, cloud computing, 50
 - loopback addresses, IPv6, 368
 - loops
 - closed-loop functioning, IoT, 51
 - Layer 2 security, 317–318
 - LoRaWAN (Long Range Wide Area Network), 54
 - low-bandwidth attacks, IPS/IDS, 58
 - LRWPAN (Long Range Wireless Personal Area Network), 54
- ## M
-
- MAB (MAC Authorization Bypass), 188, 203
 - MAC (Mandatory Access Controls), 47, 168
 - MAC addresses
 - dynamic malware analysis, 29
 - flooding, data plane (NFP), 338
 - macro infections, 16
 - malware, 12
 - AMP, 582, 637
 - AMP for Endpoints, 637, 638–639
 - AnyConnect AMP Enabler*, 650
 - Application Control*, 644–645
 - connectors*, 648
 - engines*, 650
 - exclusion sets*, 645–647
 - IP blacklists/whitelists*, 643–644
 - Outbreak Control*, 639–643
 - policies*, 648–649
 - reports*, 651–654
 - BIOS infections, 16
 - Bluetooth, 17

- crypters, 22
- distribution types, 22
- droppers, 22, 26
- dynamic analysis, 27, 28–29
- file infections, 16
- Flame, 17
- IRP, 29
- master boot record infections, 16
- packers, 22
- payloads, 17–18
- static analysis, 27–28, 29
- wrappers, 22
- management plane, traditional networks, 109**
- management plane (NFP), 333–334**
 - AAA, 339
 - availability checks, 341
 - best practices, 334–336, 339–341
 - console cables, 339
 - encrypted management protocols, 340
 - logging, 340
 - management traffic security, 338–339
 - monitoring, 340
 - NTP, 340
 - OOB management, 340, 341
 - passwords, 339–340, 341, 347–348
 - RBAC, 340
 - security, 334–336
 - user authentication, 339
- management traffic**
 - management plane (NFP), 338–339
 - security, 338–339
- man-in-the-browser attacks, 34**
- man-in-the-middle attacks, 34**
 - cloud computing, 50
 - IPv4, 374
 - IPv6, 374
- manual NAT, 443**
- Mariposa, 13**
- master boot record infections, 16**
- MD5 (Message Digest 5)**
 - authentication
 - EIGRP authentication, 384–385*
 - OSPF authentication, 383–384*
 - checksums, 85
 - HMAC, 86
 - md5sum Linux command, 85
 - secure routing protocols, 383
 - verify md5 command, 84
- MDA (Mail Delivery Agents), 619**
- MDM (Mobile Device Management) and Meraki SM, 653–654**
- mediators (IPFIX), 239**
- memory cards, 158**
- Meraki, 167**
 - Meraki SM and MDM, 653–654
 - Stealthwatch Cloud, 256
- messaging, Trojans**
 - IM, 20
 - SMS messages, 21
- metrics collection/exportation, AVC, 242**
- mGRE (multipoint) tunnels, site-to-site VPN router configurations, 486**
- micro-segmentation, 115–116, 120, 289–290, 570–571**
- microservices, 570–571**
- minikube start command, 566–567**
- minimizing control plane traffic, 379–380**
- misconfigurations, 9**
- MITRE, 9–10, 455**
- MMTF (Multi-Mode Transparent Firewalls), 416**
- mobile apps (impersonated), Trojans, 21**

mobile devices

- AnyConnect Secure Mobility, 478–479
- MDM and Meraki SM, 653–654

monitoring

- AppDynamics Cloud Monitoring, 590–593
- management plane (NFP), 340
- passive monitoring mode, NGFW/NGIPS, 420–422
- MPF (Modular Policy Frameworks), 433
- MSA (Mail Submission Agents), 619
- MSSP (Managed Security Service Providers), 73
- MTA (Mail Transfer Agents), 619
- MUA (Mail User Agents), 619
- Multi-Auth mode (802.1X), 203
- multicast addresses, 369
 - all-nodes multicast addresses, 368
 - all-routers multicast addresses, 368
 - solicited-node multicast addresses, 369
- multifactor authentication, 159, 160–161, 166, 341
- multilayer authentication, 47
- multipartite viruses, 16
- Mutiny Fuzzing Framework, 573
- MX (Mail Exchanger) records, 620

N

- nameif command, 412, 432
- NAT (Network Address Translation), 437–438, 443
 - ASA support, 396
 - auto-NAT, 443
 - dynamic NAT, 438, 441–442
 - identity NAT, 442

- manual NAT, 443
- policy NAT, 442
- remote-access VPN ASA configurations, 514
- site-to-site VPN configurations, 508–509
- static NAT, 438, 441
- TCP Intercept, 443
- NATAS virus, 16
- native VLAN, trunking, 315–316
- NAT-T (NAT-Traversal), 474
 - DMVPN, 487
 - site-to-site VPN configurations, 510
- natural disasters, 12
- NBAR2 libraries, application policies (DNA), 127
- NDP (Network Discovery Protocol), IPv6, 375
- need to know, authorization, 168
- neighbor cache resource starvation, IPv6, 375
- neighbor discovery, SeND, 377
- NETCONF, 141–143
- NetFlow, 225–227
 - anomaly detection, 229–231
 - best practices, 268–269
 - caches, 228–229
 - collection considerations, 268–269
 - configurations, 269–270
 - data leak detection/prevention, 231
 - DDoS attack mitigation, 229–231
 - deployment scenarios, 242–243
 - data center*, 246–248
 - Internet edge*, 245
 - remote VPN*, 248–249
 - site-to-site VPN*, 248–249
 - user access layer*, 243
 - WLAN, 244

- Flexible NetFlow, 228
 - application tracking (simultaneous)*, 270–271
 - configurations*, 275–285
 - flow exporters*, 275, 280–282
 - flow monitors*, 275, 282–283
 - flow samplers*, 275
 - IPFIX export format*, 283
 - key fields*, 271–273
 - non-key fields*, 273–274
 - records*, 271, 274–278
- fps, determining, 269
- incident response, 231–236
- IP accounting versus, 229
- network planning, 236
- network security, 229
- network visibility, 229
- NSEL, 248
- NX-OS configurations, 283–285
- PDU, 228
- random-sampled NetFlow, 269
- role of, 229
- scalability, 269
- threat hunting, 231–236
- timers, 284–285
- traffic engineering, 236
- versions of, 237
- Netmaster**, 120
- Netplugin**, 120
- networks**. *See also* SDN; VLAN; VPN
 - 6LoWPAN, IoT, 54
 - ACI, 110–112
 - ACL, 180
 - APIC, 110, 111–112
 - control plane, 109
 - data plane, 109
 - DNA
 - architecture of*, 121
 - policies*, 123–127
 - DNAC, 121–124
 - API, 130, 132
 - Assurance solution, 128–129
 - multivendor support, 132
 - Security solution, 132
 - enforcers, networks as, 226–227
 - infrastructure access controls, 170
 - infrastructure device images, security, 364–365
 - IoT and network resource preservation, 51
 - LAN
 - EAPoL, 179
 - VXLAN, 110, 112–114
 - LoRaWAN, IoT, 54
 - LRWPAN, IoT, 54
 - management plane, 109
 - managing traditional solutions, 109. *See also* SDN
 - network device API, 139
 - network preprocessors, 451
 - NVF, 118
 - architecture of*, 119
 - NVF MANO, 119–120
 - OPNFV, 118–119
 - overlays, 112–114
 - OVN, 117
 - P2P networks, Trojans, 20
 - PAN
 - 6LoWPAN, 54
 - LRWPAN, 54
 - planning, NetFlow, 236
 - programmability
 - API, 132–133, 136–140
 - DevNet, 136, 142
 - gNMI, 145–146

- NETCONF, 141–143
- OpenConfig, 145–146
- Python programming, 133–136
- RESTCONF, 143–145
- YANG models, 139–140
- security forensics, NetFlow, 231–236
- segmentation, 285
 - application-based segmentation, 288–289
 - data-driven segmentation, 286–288
 - ISE, 290–291
 - micro-segmentation, 289–290
 - SGT assignments/deployments, 294
 - SXP, 292–294
- sensors, networks as, 226–227
- visibility, 224–225
 - AVC, 241–242
 - CTA, 262–268
 - enforcers, networks as, 226–227
 - ETA, 262
 - five-tuples, 227
 - flow, defined, 227
 - flow, sessions versus, 229
 - NetFlow, 225–237, 242–243, 268–285
 - NVM, 249
 - sensors, networks as, 226–227
 - Stealthwatch, 230–231, 233, 243, 250–261
- visibility (networks), IPFIX, 237–241, 283
- VXLAN, 110, 112–114, 120
- WAN, LoRaWAN, 54
- WLAN, NetFlow deployment scenario, 244
- Neutron, 117
- next-generation encryption protocols, 89–90
- NFP (Network Foundation Protection), 332
 - auto secure command-line utility, 334
 - control plane, 333–334
 - best practices, 336–337
 - CoPP, 336
 - CPPr, 336–337
 - minimizing traffic, 379–380
 - secure routing protocols, 379
 - security, 336–337
 - SPD, 337
 - data plane, 333–334
 - best practices, 337–338
 - IPv6 configuration/security, 365–378
 - security, 337–338
 - framework of, 333
 - implementing, 333–334
 - importance of, 332
 - interdependence, 333
 - management plane, 333–334
 - AAA, 339
 - availability checks, 341
 - best practices, 334–336, 339–341
 - console cables, 339
 - encrypted management protocols, 340
 - logging, 340
 - Login Password Retry Lockout, 339–340
 - management traffic security, 338–339
 - monitoring, 340
 - NTP, 340
 - passwords, 339, 341, 347–348
 - RBAC, 340

- security*, 333–334
 - user authentication*, 339
 - passwords, 341
 - NGFW (Next-Generation Firewalls), 395–396**
 - Firepower, 398
 - 1000 series*, 397
 - 2100 series*, 397–398
 - 4100 series*, 398
 - 9300 series*, 399
 - inline pairs, 420–421
 - passive ERSPAN mode, 422
 - passive (monitoring) mode, 420–422
 - NGIPS (Next-Generation Intrusion Prevention Systems), 395–396, 399–401**
 - FMC, 401–404
 - inline pairs, 420–421
 - passive ERSPAN mode, 422
 - passive (monitoring) mode, 420–422
 - preprocessors, 450–452
 - variables, 449–450
 - NHRP (Next Hop Resolution Protocol), 486–487**
 - NIST (National Institute of Standards and Technology)**
 - cybersecurity framework, 7–8
 - 1800 Series*, 8
 - FIPS*, 7
 - ITL*, 8
 - NISTIR*, 8
 - SP 800 Series*, 7
 - incident response, 55–56
 - IRP, 60
 - SOP, defined, 60–61
 - SP 500–292, 48, 552
 - SP 800–52 revision 2, 91
 - SP 800–61, 61
 - SP 800–61 revision 2, 55, 60, 62–63, 231
 - SP 800–63B, 157
 - SP 800–145, 47–48
 - NISTIR (NIST Internal or Interagency Reports), 8**
 - Nomad, 560**
 - nonlocal multicast addresses, filtering, 377
 - northbound API, 118
 - NSEL (NetFlow Secure Event Logging), 248**
 - NTP (Network Time Protocol), 346**
 - configurations, 363–364
 - management plane (NFP), 335, 340
 - NTPv3, authentication keys, 363–364
 - synchronization, verifying, 364
 - NVD (National Vulnerabilities Database), 30**
 - NVF (Network Function Virtualization), 118**
 - architecture of, 119
 - NVF MANO, 119–120
 - OPNFV, 118, 119
 - NVF MANO (NVF Management and Network Orchestration), 119–120**
 - NVM (Network Visibility Module), 249**
 - NX-OS, 346–349**
 - Nyeta, 23**
-
- O**
- OASIS, 455**
 - object capability, authorization, 167
 - object grouping, ACL, 435–436
 - OCI (Open Container Initiative), 561**
 - OCSP (Online Certificate Status Protocol), 100**
 - ODL (OpenDaylight), 117–118**

Office 365, email security, 583–584

OllyDbg, 28

one-to-one address mapping, 438

OOB (Out-of-Band) management, management plane (NFP), 340–341

OpenAPI, Swagger, 39

Open Authentication, 204

OpenC2 (Open Command and Control), 15

OpenConfig, network programmability, 145–146

OpenDNS, Umbrella, 577
architecture of, 577–578
Investigate, 580–582
SIG, 578–580

OpenIOC (Open Indicators of Compromise), 15

open source software, vulnerabilities, 40

OPNFV (Open Platform for Network Function Virtualization), 118–119

option templates (IPFIX), 241

organized crime, 13

OS (Operating Systems)
NX-OS, 346–349
vulnerabilities, 9

OSPF (Open Shortest Path First)
ip ospf authentication-key command, 383
ip ospf message-digest-key command, 383
MD5 authentication, 383–384
OSPF over IPsec, 509

OTP (One-Time Pads), 81–82
OTP (One-Time Passwords), 157–158

Outbreak Control, 639–643

out-of-band authentication, 47, 158

out-of-band SQL injection, 32

overlays (network), 112–114

OVN (Open Virtual Network), 117

OVS (Open vSwitch), 110, 117

OVSDB (Open vSwitch Database), 110, 117

OWASP (Open Web Application Security Project)
Proactive Controls, 571–572
Top 10 list, 40

ownership, authentication by, 157–158

P

P2P (Peer-to-Peer) networks, Trojans, 20

PaaS (Platform as a Service), 48, 552

packers, malware distribution, 22

packet amplification attacks, 376

PAN (Personal Area Networks), 54

parser views
creating, 358–359
RBAC, 358–360
user accounts, associating with views, 360

partitioning firewalls, 414

PAS. *See* ISO

passive ERSPAN mode, NGFW/NGIPS, 422

passive (monitoring) mode, NGFW/NGIPS, 420–422

passwords, 156–157
cracking, 33–34
Login Password Retry Lockout, management plane (NFP), 339–340
management plane (NFP), 339, 347–348
multifactor authentication, 341
NFP, 341
OTP, 157–158
pxGrid, 184

- security passwords min-length command, 341
- single-factor authentication, 341
- Trojans, 20
- PAT (Port Address Translation)**
 - ASA and, 440
 - dynamic PAT, 442
 - policy PAT, 442
 - static PAT, 441
- patches**
 - cloud patch management, 575
 - Firepower, 458
- pattern change evasion, IDS/IPS, 58**
- payloads (viruses), 18**
- PDU (Protocol Data Units), 228**
- Peach, 573**
- Pearson Cert Practice Test engine, exam preparation, 659**
- persistent (stored) XSS attacks, 36**
- PFS (Perfect Forward Secrecy), site-to-site VPN configurations, 509**
- PGP (Pretty Good Privacy), key servers, 93**
- physical access, Trojans, 21**
- physical security IPv4/IPv6, 372**
- ping command, 24**
- Ping of Death, 13**
- PKCS (Public Key Cryptography Standards), 83, 99**
- PKI (Public Key Infrastructure), 87, 93**
 - cross-certifying CA, 102
 - hierarchical CA, 101–102
 - single root CA, 101
 - subordinate CA, 101
 - topologies, 101–102
- planning networks, NetFlow, 236**
- platform settings policies, Firepower, 450**
- playbooks, incident response, 63–64**
- POC exploits, 10**
- poison apple attacks/USB key drops, 19**
- policy NAT and ASA, 442**
- policy PAT and ASA, 442**
- polyalphabetic method, ciphers, 81**
- polymorphic viruses, 17**
- POP (Post Office Protocol), 620**
- PortFast configurations, 321–322**
- ports**
 - access lists, IPv6, 377
 - Layer 2 security, 324, 325–327
 - MAC address flooding, 338
 - PAT, 440–441
 - redirecting, 441
 - security, 802.1X, 203
 - STP
 - new ports, 321*
 - port states, 321*
 - switch ports, locking down, 323
 - Trojan ports, 19
 - VLAN, locking down switch ports, 323
- possession, authentication by, 157–158**
- POST (Power On Self-Tests), BIOS infections, 16**
- post-incident activity phase (IRP), 63**
- posture assessments, 192–193**
- practice tests, Pearson Cert Practice Test engine, 659**
- predicting session tokens, 34**
- preparation phase (IRP), 61**
- preparing for exams, 658**
 - Blueprints, 658
 - DIKTA questions, 658
 - hands-on activities, 658
 - Pearson Cert Practice Test engine, 659

- “Review Questions” sections, 659
- review/study plans, 658–659
- preprocessors (NGIPS), 450–452
- private clouds, 48, 552
- private key cryptography, 83, 93
- privileges
 - access control management, 45–46
 - custom privileges, 344
 - principle of least privilege, 155
 - RBAC privilege levels, 356–358
- profiling services (ISE), 184–187
- programmability (networks)
 - API, 132–133, 136
 - GraphQL*, 137
 - network device API*, 139
 - REST*, 137–139
 - SOAP*, 136, 137
 - Swagger (OpenAPI)*, 137
 - WADL documents*, 137
 - WSDL documents*, 137
 - YANG models*, 139–140
 - DevNet, 136, 142
 - gNMI, 145–146
 - NETCONF, 141–143
 - OpenConfig, 145–146
 - Python programming, 133–136
 - RESTCONF, 143–145
 - YANG models, 139–140
- providers (cloud computing)
 - liability, 50
 - responsibilities, 573–575
- proxies
 - SOCKS proxies, 607–608
 - WSA, 605–606
- proxy Trojans, 19
- proxying addresses, IDS/IPS, 58
- PSIRT (Product Security Incident Response Teams), 66–67, 70

- PTEP (Physical Tunnel Endpoint)
 - function, 110
- public clouds, 48, 552
- public key cryptography, 83
 - digital certificates, 97
 - PKCS, 99
 - public key pairs, 93
 - root certificates, 96
- push protocols, 238
- pxGrid (Platform Exchange Grid), 182–184
- Pyeta, 23
- Python programming, 133–136

Q

- quantum computing, cryptography, 86
- queries, SQLi attacks, 32
- questions, exam preparation
 - DIKTA questions, 658
 - “Review Questions” sections, 659

R

- race conditions, 38
- Radamsa, 573
- RADIUS (Remote Authentication Dial-In User Service), 173–176, 179, 187
 - authentication configuration, 202–205
 - client-based remote access SSL VPN, 526
- rainbow tables, authentication-based vulnerabilities, 33–34
- random-sampled NetFlow, 269
- ransomware (data hiding), 19, 23
- RAT (Remote Access Trojans), 18
- RAT (Recipient Access Tables), 622

- RBAC (Role-Based Access Controls), 47, 168–169, 344
 - management plane (NFP), 334–335, 340
 - parser views, 358–360
 - privilege levels, 356–358
- recovery (disaster), cloud computing, 50
- redirecting ports, 441
- reflected DoS attacks, availability (CIA triad), 45
- reflected XSS attacks, 36
- registries (containers), 561
- regulatory requirements, cloud computing, 49
- remote VPN, NetFlow deployment scenario, 248–249
- remote-access VPN, 468–469
 - ASA firewall configurations, 511–512
 - attributes*, 518
 - client-based remote access SSL VPN*, 524–526
 - clientless remote access SSL VPN*, 514–515
 - design considerations*, 515–516
 - group policy*, 513
 - IP pools*, 513
 - IPsec*, 512–514
 - NAT exemptions*, 514
 - policy inheritance model*, 518
 - tunnel groups*, 513–514
 - FTD, 530–531, 540
 - Policy Wizard, 531–540
- reporting incidents, incident response, 58–59
- residual risk, defined, 12
- REST (Representational State Transfer), 38, 137–139
- RESTCONF, 143–145
- RESTful API, IoT, 53
- ret2libc (return-to-libc) attacks, 39
- retrospection, 456–457
- “Review Questions” sections, exam preparation, 659
- review/study plans, exam preparation, 658–659
- revoking digital certificates, 98, 99–100
- RFC 2409, 470
- RFC 3547, 489
- RFC 4594, 127
- RFC 5585, 623
- RFC 5617, 623
- RFC 5863, 623
- RFC 5996, 470
- RFC 6241, 141
- RFC 6242, 141
- RFC 6347, 529
- RFC 6376, 623
- RFC 6526, 241
- RH0 packets, IPv6, 377
- riding sessions, cloud computing, 50
- RIP (Routing Information Protocol)
 - MD5 authentication, 385–386
 - routing update authentication, 385–386
- risk
 - defined, 12
 - residual risk, 12
- rogue IPv6 devices, 377
- root certificates, 95–96
- Root Guard, 324, 325
- routed firewalls, 413
- routing
 - all-routers multicast addresses, 368
 - IPv4
 - routing attacks*, 374
 - routing protocol security*, 373

- IPv6, 370–372
 - routing attacks*, 374
 - routing protocol security*, 373
 - router access authentication, 342–343
 - router-on-a-stick, 316–317
 - RRI, site-to-site VPN configurations, 509
 - secure routing protocols, 379, 383
 - site-to-site VPN configurations, 479
 - debug commands*, 496–502
 - DMVPN*, 486–489
 - FlexVPN*, 492–496, 499–501
 - GETVPN*, 489–492
 - GRE over IPsec*, 482–484
 - IOS/IOS-XE devices*, 479–482
 - mGRE tunnels*, 486
 - R1 configurations*, 480–481
 - R2 configurations*, 481–482
 - show commands*, 496–502
 - topologies*, 480
 - troubleshooting IPsec tunnels*, 496–502
 - tunnel interfaces*, 482, 484–486
 - TACACS+, debugging in routers, 199–200
 - troubleshooting, AAA with CLI, 353–356
 - update authentication
 - BGP*, 386–387
 - RIP*, 385–386
 - VLAN, inter-VLAN routing, 316–317
 - RPC (Remote Procedure Calls), 145
 - RPF (Reverse Path Forwarding), Unicast RPF, 380
 - RRI (Reverse Route Injection), site-to-site VPN configurations, 509
 - RSA (Rivest-Shamir-Adleman), 83, 93–94
 - rsa-signatures. *See* digital signatures
 - RSTP (Rapid Spanning Tree Protocol), 321–322
 - rule-based access control, 169
-
- ## S
-
- SaaS (Software as a Service), 48, 251–256, 552
 - same-security-traffic permit inter-interface command, 412
 - SAML (Security Assertion Markup Language), 159, 165
 - SamSam, 23
 - sandboxes
 - malware analysis, 29
 - ThreatGrid, 29
 - SAST (Static Application Security Testing), 572–573
 - SCADA preprocessors, 451
 - scalability, NetFlow, 269
 - SCEP (Simple Certificate Enrollment Protocol), 99
 - scripting
 - script kiddies, 13
 - XSS, 32
 - Scrum framework, agile development methodology (cloud computing), 554–555
 - SCTP (Stream Control Transmission Protocol), 241
 - SDL (Secure Development Life Cycle), 70–71
 - SDLC (Secure Development Life Cycle), 555
 - SDN (Software-Defined Networking), 108–109
 - ACI, 110–112
 - APIC, 110, 111–112
 - Contiv, 120

- controllers, 110
- DNA
 - architecture of*, 121
 - policies*, 123–127
- DNAC, 121–124
 - API*, 130, 132
- Assurance solution, 128–129
- multivendor support, 132
- Security solution, 132
- micro-segmentation, 115–116, 120
- network overlays, 112–114
- Neutron, 117
- northbound API, 118
- NVF, 118
 - architecture of*, 119
 - NVF MANO*, 119–120
 - OPNFV*, 118, 119
- ODL, 117–118
- open-source initiatives, 117–118
- OVN, 117
- OVS, 110, 117
- OVSDB, 110, 117
- southbound API, 118
- VTEP, 110–111
- VXLAN, 110, 112–114
- SD-WAN configurations, ZBFW, 411–412
- search routines, 17
- searchsploits, 10–11
- secure routing protocols, control plane (NFP), 379
- secure system files, management plane (NFP), 341
- security contexts, firewalls, 414
- security intelligence, updating, 457–458
- security labels, authorization, 167
- security passwords min-length command, 341
- security plans, IPv6, 372
- security policies, IPv4/IPv6, 373
- Security Score (Tetration), 595
- security-software disablers, 19
- Security solution, DNAC, 132
- security zones, 406–407
- segmentation (networks), 285
 - application-based segmentation, 288–289
 - data-driven segmentation, 286–288
 - ISE, 290–291
 - micro-segmentation, 289–290
 - SGT assignments/deployments, 294
 - SXP, 292–294
- SEI, CERT SEI, 72
- SeND (Secure Neighbor Discovery), IPv6, 377
- SenderBase, 622
- sensors, networks as, 226–227
- separation of duties, 155
- sequence numbers (TCP), 25
- serial numbers
 - digital certificates, 97
 - root certificates, 95
- serverless cloud computing, 559
- servers (PGP key), 93
- service timestamps, syslog, 363
- session
- sessions
 - flow versus, 229
 - hijacking, 34, 50
 - riding, cloud computing, 50
 - sniffing, 34
 - token predictions, 34
- SGACL (Security Group-based ACL), 181
- SGT (Security Group Tags), 188, 294

- SHA (Secure Hash Algorithm)
 - checksums, 85
 - HMAC, 86
- SHA512 checksum, 84
 - shasum Linux command, 85
 - vulnerabilities, 85–86
- sharing information/coordination,
 - incident response, 64
- shortcuts, IPv6 addresses, 367
- show commands
 - IPsec tunnels, troubleshooting, 496–502
 - site-to-site VPN router configurations, 496–502
- show crypto ikev2 sa command, 498
- show crypto ikev2 sa detailed command, 498
- show crypto ikev2 sa session command, 498–499
- show crypto isakmp sa command, 498
- show flow exporter command, 281
- show flow monitor command, 279
- show flow monitor name NY-ASR-FLOW-MON-1 cache record format command, 281–282
- show flow record command, 278
- show interface trunk command, 314
- show interfaces Gi0/2 switchport command, 313, 314–315
- show ip cef output command, 379–380
- show ipv6 route command, 372
- show monitor event-trace crypto ikev2 command, 501
- show monitor event-trace crypto ikev2 error all command, 502
- show policy-map control-plane command, 380
- show running-config flow exporter command, 281
- show running-config flow monitor command, 279–280
- show running-config flow record command, 278
- show vlan brief command, 312
- show vlan id command, 312–313
- show-access list command, 435
- shrinkwrap software, vulnerabilities, 9
- shutdowns (four-step), TCP, 25
- side-channel attacks, cloud computing, 51
- SIG (Security Internet Gateway), 578–580
- signatures (digital), 86–89
 - digital certificates, 97
 - DSA, 84
 - RSA, 93–94
- single root CA, 101
- single-factor authentication, 47, 159, 341
- SIP preprocessors, 451
- site-to-site VPN, 468–469
 - ASA firewall configurations, 502–503
 - bypass NAT*, 508–509
 - crypto maps*, 506–508
 - fragmentation*, 510–511
 - IPsec policies*, 505–506
 - ISAKMP*, 503–504
 - management access*, 510
 - NAT-T*, 510
 - OSPF over IPsec*, 509
 - PFS*, 509
 - traffic-filtering*, 503–508
 - tunnel default gateways*, 510
 - tunnel groups*, 504–505
 - FTD, 541–543
 - NetFlow deployment scenario, 248–249
 - router configurations, 479
 - debug commands*, 496–502
 - DMVPN*, 486–489

- FlexVPN*, 492–496, 499–501
- GETVPN*, 489–492
- GRE over IPsec*, 482–484
- IOS/IOS-XE devices*, 479–482
- mGRE tunnels*, 486
- R1 configurations*, 480–481
- R2 configurations*, 481–482
- show commands*, 496–502
- topologies*, 480
- troubleshooting IPsec tunnels*, 496–502
- tunnel interfaces*, 482, 484–486
- sizing ISE distributed deployments, 214
- SKEYID, 472
- SLA (Service Level Agreements), 49
- smartcards, 158
- SMC (Stealthwatch Management Console), 250
- SMS messages, Trojans, 21
- SMTF (Single-Mode Transparent Firewalls), 414–416
- SMTP (Simple Mail Transfer Protocol)
 - ACL and SMTP traffic, 432–433
 - ESA, 623
 - preprocessors, 451
- sniffing attacks, IPv4/IPv6, 374
- SNMP, management plane (NFP), 335
- SOAP (Simple Object Access Protocol), 38, 136, 137
- SOCKS proxies, 607–608
- Sodinokibi, 23
- software
 - assurance tools/methods, 572–573
 - DAST, 572–573
 - Findsebugs, 572–573
 - fuzz testing (fuzzing), 573
 - open source software, vulnerabilities, 40
 - SAST, 572–573
 - shrinkwrap software, vulnerabilities, 9
 - SonarQube, 573
 - updates, 458
 - vulnerabilities
 - authentication-based vulnerabilities*, 32–35
 - buffer overflows*, 39
 - cookie manipulation attacks*, 37–38
 - CVE, 30
 - injection vulnerabilities*, 30–32
 - NVD, 30
 - open source software*, 40
 - OWASP Top 10 list*, 40
 - race conditions*, 38
 - ret2libc attacks*, 39
 - unprotected API*, 38–39
 - XSRF, 37
 - XSS, 35–37
- solicited-node multicast addresses, IPv6, 369
- SonarQube, 573
- SOP (Standard Operating Procedures), 60–61
- southbound API, 118
- SP (Special Publication)
 - 500–292, 48, 552
 - 800 Series, 7
 - 800–52 revision 2, 91
 - 800–61, 56, 61
 - 800–61 revision 2, 55, 60, 62–63, 231
 - 800–63B, 157
 - 800–83, 55
 - 800–86, 55
 - 800–145, 47–48
 - 1800 Series, 8

- sparse infections**, 17
- SPD (Selective Packet Discard)**, 337
- Spero**, 454, 650
- SPF (Sender Policy Framework)**, 583, 623
- spine nodes/switches**, 111
- split tunneling, AnyConnect Secure Mobility**, 528–529
- spoke configurations, DMVPN**, 488–489
- spoofing addresses, IDS/IPS**, 58
- spoofing attacks**
 - ARP spoofing, 330, 338
 - data plane (NFP), 338
 - IPv4, 374
 - IPv6, 374
- Spora**, 23
- spyware**, 16, 26
 - advertising, 26
 - droppers, 26
 - surveillance, 26
- SQL injection, cloud computing**, 50
- SQLi (SQL injection)**, 30, 31–32
 - blind (inferential) SQL injection, 32
 - example of, 31
 - in-band SQL injection, 32
 - out-of-band SQL injection, 32
 - queries, 32
 - SQL statements, 30–31
- SRU (Snort Rules Updates)**, 458
- SSH (Secure Shell)**, 360–362, 451
- SSL (Secure Socket Layer)**, 91
 - preprocessors, 451
 - VPN, 476–479
 - application access*, 524–525
 - client-based remote access SSL VPN*, 524–526
 - configurations*, 516–518
 - enabling*, 522–523
 - group policies*, 518–519
 - tunnel groups*, 519–520
 - user authentication*, 520–522
 - Webtype ACL*, 523–524
- SSO (Single Sign-On) applications**, 164–167
 - Duo Security, 166
 - SAML, 159
- stacks (dual), IPv6**, 376
- standard ACL**, 430, 435–436
- state-sponsored/government threats**, 13
- static malware analysis**, 27–28, 29
 - BinText, 27
 - edb, 27
 - Ghidra, 28
 - IDA Pro, 27
 - OllyDbg, 28
 - UPX, 27
- static NAT and ASA**, 438, 441
- static PAT and ASA**, 441
- stealth AnyConnect, posture assessments**, 193
- Stealthwatch**, 132, 230–231, 243, 250
 - components of, 250–251
 - flow licenses, 250
 - Flow Sensor, 233
 - FlowCollector, 250
 - FlowReplicator, 251
 - FlowSensor, 251
 - on-premises appliances, 256–259
 - SMC, 250
 - Stealthwatch Cloud, 251–256, 590
 - threat hunting, 258–261
- STIX (Structured Threat Information EXpression)**, 15, 455
- storage (data), Trojans**, 20

stored (persistent) XSS attacks, 36
 Storm, 13
 Storm Control, 324
 STP (Spanning Tree Protocol), 317–318

- annotations, 318–320
- ports
 - new ports*, 321
 - port states*, 321
- Root Guard, 325
- RSTP, configurations, 321–322
- time until forwarding, 321–322
- verification, 318–320

 stream ciphers, 82
 Stuxnet, 12
 subjects, digital certificates, 97
 subordinate CA, 101
 substitution method, ciphers, 81
 Sun RPC preprocessors, 450
 surveillance, spyware, 26
 Swagger (OpenAPI), 39, 137
 switches

- leaf switches, 110–111
- ToR switches, 111

 SXP (Scalable Group Tag Exchange Protocol), 292–294
 symmetric encryption algorithms, 82–83
 SYN packets, 25
 syslog

- configurations, 362–363
- logging files, 345–346
- management plane (NFP), 335
- service timestamps, 363
- severity levels, 346

 sysopt connection permit-vpn command, 508
 system root CA certificates, 88–89

T

tabletop exercises/playbooks, incident response, 63–64
 TACACS+ (Terminal Access Control Access Control System Plus), 174–176

- access configuration, 196–199, 200–202
- debugging, 199–200

 Talos, 458

- AMP and, 453
- email security, 582

 TAN grabbers, 19
 TAXII (Trusted Automated EXchange of Indicator Information), 15, 455
 TC-NAC, CoA, 193
 TCP (Transmission Control Protocol)

- ACK packets, 25
- acknowledgements, 25
- control information exchanges, 24
- covert communication, 24–25
- four-step shutdowns, 25
- process of, 24–25
- sequence numbers, 25
- SYN packets, 25
- TCP Intercept, 443
- three-step handshakes, 24

 Teardrop, 13
 Telnet

- encrypted management protocols, 344–345
- FTP and Telnet preprocessors, 450

 templates (IPFIX), 238

- example of, 240
- option templates, 241
- structure of, 239–240

 temporal agents, posture assessments, 192

- terminal monitor command, 199**
- terrorist groups, 13**
- test aaa command, 356**
- testing**
 - fuzz testing (fuzzing), 573
 - Pearson Cert Practice Test engine, 659
 - penetration testing, CSP, 575–577
 - XSS, 37
- TETRA, AMP for Endpoints, 650**
- Tetration, 593–594**
 - ADM, 594
 - connectors, 595
 - Forensics feature, 594
 - Security Dashboard, 594–595
 - Security Score, 595
 - Vulnerability Dashboard, 595–596
- Threat Grid, 29, 452–453, 455–456**
- threat hunting, NetFlow, 231–236**
- threats**
 - catastrophic damage, 12
 - covert communication, 23–25
 - CTA, 262–268
 - cyberattacks, 12
 - cybersecurity threats, defined, 9
 - DDoS attacks, 13
 - defined, 12
 - disclosure of confidential information, 12–13
 - DoS attacks, 13
 - hacker attacks, 12
 - hacktivists, 13
 - IPv4 common threats, 373–374
 - IPv6 common threats, 373–374
 - IRP, 29
 - keyloggers, 25–26
 - Layer 2 security, 322–323
 - malware, 12
 - distribution types, 22*
 - dynamic analysis, 27–29*
 - payloads, 17–18*
 - static analysis, 27–29*
 - transmission methods, 16–17*
 - Mariposa, 13
 - natural disasters, 12
 - Ping of Death, 13
 - ransomware (data hiding), 19, 23
 - spyware, 16, 26–27
 - Stealthwatch threat hunting, 258–261
 - Storm, 13
 - Stuxnet, 12
 - Teardrop, 13
 - threat actors
 - crackers, 13*
 - defined, 13–14*
 - hackers, 13–14*
 - organized crime, 13*
 - script kiddies, 13*
 - state-sponsored/government threats, 13*
 - terrorist groups, 13*
 - threat detection preprocessors, 451
 - threat intelligence
 - CyBOX, 15*
 - defined, 14*
 - OpenC2, 15*
 - OpenIOC, 15*
 - process of, 14*
 - standards, 14–15*
 - STIX, 15, 455*
 - TAXII, 15, 455*
 - updating, 457–458*
 - Threat Response, 654–655

- Trojans
 - communication methods*, 19
 - defined*, 18
 - effects of*, 22
 - goals of*, 20
 - infection mechanisms*, 20–21
 - ports*, 19
 - types of*, 18–19
- viruses, 12, 16
 - components of*, 17–18
 - transmission methods*, 16–17
 - types of*, 16–17
- weather-related threats, 12
- worms, 16
 - transmission methods*, 16–17
 - types of*, 16–17
- three-step handshakes (TCP), 24
- through-the-box traffic filtering, 431
- thumbprint algorithms
 - digital certificates, 98
 - root certificates, 96
- time until forwarding, STP, 321–322
- timers (NetFlow), 284–285
- timestamps (service), syslog, 363
- TLS (Transport Layer Security), 91
- TOCTOU attacks. *See* race conditions
- tokens, session token predictions, 34
- ToR (Top-of-Rack) switches, 111
- to-the-box traffic filtering, 434–435
- TPS security, 71
- tracking IPv6 devices, 377
- traffic copy policy (DNA), 127
- traffic engineering, NetFlow, 236
- traffic-filtering
 - ASA, 396–397
 - through-the-box traffic filtering, 431
 - to-the-box traffic filtering, 434–435
- training, cloud computing, 49
- transform sets, 479
- transmission methods of malware, 16–17
- transparent firewalls, 413–414
 - MMTF, 416
 - SMTF, 414–416
- transparent mode (WSA), 608–609
- transposition method, ciphers, 81
- trigger routines, 18
- Trojans
 - APT, 20
 - backdoors, 19
 - browser/brower extension
 - vulnerabilities, 21
 - communication methods, 19
 - credit card data, 20
 - data hiding (ransomware), 19
 - data storage, 20
 - defined, 18
 - DoS attacks, 19
 - e-banking, 19
 - effects of, 22
 - electronic/digital wallets, 20
 - email attachments, 21
 - freeware, 21
 - FTP Trojans, 19
 - goals of, 20
 - IM, 20
 - impersonated mobile apps, 21
 - infection mechanisms, 20–21
 - insider information, 20
 - IRC, 21
 - P2P networks, 20
 - passwords, 20
 - physical access, 21
 - poison apple attacks/USB key
 - drops, 19
 - ports, 19
 - proxy Trojans, 19

- RAT, 18
- security-software disablers, 19
- SMS messages, 21
- TAN grabbers, 19
- types of, 18–19
- watering holes, 21
- Zeus, 19
- troubleshooting**
 - AAA for Cisco routers, 353–356
 - IPsec tunnels, site-to-site VPN configurations, 496–502
 - remote-access VPN, 540
- true positives/negatives, incident response, 57–58**
- trunking, VLAN, 313–314**
 - 802.1Q trunking, 313–315
 - broadcast frames, 315
 - interfaces as trunk ports, 313–315
 - native VLAN, 315–316
 - port negotiations, 316
 - show interface trunk command, 314
 - show interfaces Gi0/2 switchport command, 314–315
- TrustSec, 190–192**
 - ACI integration, 298–301
 - active policy enforcement, 295–298
 - monitor mode deployments, 294–295
 - SGT, 188
- TTL (Time-To-Live), IPv6, 380**
- tunneling**
 - AnyConnect Secure Mobility, 528–529
 - application layer tunneling, 25
 - client-based remote access SSL VPN, 525–526
 - IPv6, 376, 377
 - remote-access VPN ASA configurations, 513–514
 - site-to-site VPN
 - ASA firewall configurations, 504–505
 - VPN router configurations, tunnel interfaces, 482, 484–486
 - split tunneling, 528–529
 - SSL VPN, 519–520
 - tunnel default gateways, site-to-site VPN configurations, 510
 - tunnel mode command, 485
 - tunnel mode gre multipoint command, 486
 - UDP tunneling, 25
 - VTI, 485

U

- UDP (User Datagram Protocol)**
 - (User Datagram Protocol)
 - covert communication, 25
 - dnscat, 25
 - flow exporters, 280
 - IKE, 472
 - management plane (NFP), 335
 - tunneling, 25
 - VLAN, 113
- Umbrella, 167, 577**
 - architecture of, 577–578
 - Investigate, 580–582
 - SIG, 578–580
 - Stealthwatch Cloud, 256
- unauthenticated/authenticated guess access, 188**
- unauthorized access, IPv4/IPv6, 374**
- unicast addresses, IPv6, 368–369**
- Unicast RPF (Reverse Path Forwarding), 380**
- UNIX, Duo Security, 161–162**

unprotected API, 38–39
 unreachable (ICMP), 380
 updates
 exams, 686–687
 Firepower, 458
 geolocation updates, 458
 routing update authentication
 BGP, 386–387
 RIP, 385–386
 security intelligence, 457–458
 software, 458
 SRU, 458
 threat intelligence, 457–458
 UPX (Ultimate Packer for Executables),
 27
 URL, database record retrieval, 35
 USB key drops/poison apple attacks,
 19
 US-CERT, 71, 72
 user access layer, NetFlow deployment
 scenario, 243
 user accounts, parser views, 360
 user authentication, 342
 management plane (NFP), 339,
 349–353
 SSL VPN, 520–522

V

validation, digital certificates, 97
 validity dates, root certificates, 96
 variables
 FMC, 449
 NGIPS, 449–450
 VDB (Vulnerability Database) updates,
 458
 verification
 CoPP configurations, 382
 data integrity, hashes, 84–86

 digital signatures, 87
 Kubernetes versions, 567
 STP, 318–320
 verify md5 command, 84
 VERIS community database, examples
 of data breaches, 156
 viability (long-term), cloud computing,
 50
 virtual “sub” interfaces, inter-VLAN
 routing, 316–317
 virtualization, NVE, 118
 architecture of, 119
 NVE MANO, 119–120
 OPNFV, 118, 119
 viruses, 12, 16
 antidetection routines, 18
 clusters, 16
 components of, 17–18
 fast infections, 17
 hoaxes, 17
 infection routines, 18
 macro infections, 16
 multipartite viruses, 16
 NATAS virus, 16
 payloads, 18
 polymorphic viruses, 17
 search routines, 17
 sparse infections, 17
 transmission methods, 16–17
 trigger routines, 18
 types of, 16–17
 VirusTotal website, 29
 visibility (networks), 224–225
 AVC, 241
 application recognition,
 241–242
 metrics collection/exportation,
 242

- CTA, 262–268
- enforcers, networks as, 226–227
- ETA, 262
- five-tuples, 227
- flow
 - defined*, 227
 - sessions versus*, 229
- IPFIX, 237–238
 - architecture of*, 238
 - Flexible NetFlow and IPFIX export format*, 283
 - mediators*, 239
 - SCTP, 241
 - templates*, 238, 239–241
- NetFlow, 225–227, 229
 - anomaly detection*, 229–231
 - best practices*, 268–269
 - caches*, 228–229
 - collection considerations*, 268–269
 - configurations*, 269–270
 - data leak detection/prevention*, 231
 - DDoS attack mitigation*, 229–231
 - deployment scenarios*, 242–249
 - Flexible NetFlow*, 228, 270–275, 283
 - fps, determining*, 269
 - incident response*, 231–236
 - IP accounting versus*, 229
 - network planning*, 236
 - network security*, 229
 - NSEL, 248
 - NX-OS configurations*, 283–285
 - PDU, 228
 - random-sampled NetFlow*, 269
 - role of*, 229
 - scalability*, 269
 - threat hunting*, 231–236
 - timers*, 284–285
 - traffic engineering*, 236
 - versions of*, 237
- NVM, 249
- sensors, networks as, 226–227
- Stealthwatch, 230–231, 243, 250
 - components of*, 250–251
 - Flow Sensor*, 233
 - on-premises appliances*, 256–259
 - Stealthwatch Cloud*, 251–256
 - threat hunting*, 258–261
- VLAN (Virtual Local Area Networks), 310–311
 - 802.1Q trunking, 313–315
 - ACL, 181
 - creating, 311
 - defined, 311
 - do not allow negotiations, 323
 - example of, 311
 - inter-VLAN routing, 316
 - router-on-a-stick*, 316–317
 - virtual “sub” interfaces*, 316–317
 - native VLAN, trunking, 315–316
 - show interfaces Gi0/2 switchport command, 313
 - show vlan brief command, 312
 - show vlan id command, 312–313
 - STP, instances of, 321
 - switch ports, locking down, 323
 - trunking, 313–314
 - broadcast frames*, 315
 - interfaces as trunk ports*, 314
 - native VLAN*, 315–316
 - port negotiations*, 316

- show interface trunk command*, 314
- show interfaces Gi0/2 switchport command*, 314–315
- VLAN 10 interface assignments, 312
- VLAN 20 interface assignments, 312
- VM (Virtual Machines)**
 - dynamic malware analysis, 28–29
 - KVM and ISE, 182
 - WebSploit, 562
- VMware ESXi and ISE**, 182
- VNID (VXLAN Network Identifiers)**, 113
- VPN (Virtual Private Networks)**, 454
 - AnyConnect Secure Mobility, 478–479, 527–529
 - CoA, 195–196
 - DMVPN, 486
 - example of*, 486
 - hub configurations*, 487–488
 - NAT-T*, 487
 - NHRP*, 486–487
 - spoke configurations*, 488–489
 - FlexVPN, 492–496, 499–501
 - GETVPN, 489–492
 - IPsec
 - IKE*, 470–474
 - VPN*, 499
 - RADIUS, 187
 - remote-access VPN, 468–469
 - ASA firewall configurations*, 502–540
 - FTD*, 530–531, 540
 - NetFlow deployment scenario*, 248–249
 - Policy Wizard*, 531–540
 - site-to-site VPN, 468–469
 - ASA firewall configurations*, 502–511
 - FTD*, 541–543
 - NetFlow deployment scenario*, 248–249
 - router configurations*, 479–502
 - SSL VPN, 476–479
 - application access*, 524–525
 - client-based remote access SSL VPN*, 524–526
 - configurations*, 516–518
 - enabling*, 522–523
 - group policies*, 518–519
 - tunnel groups*, 519–520
 - user authentication*, 520–522
 - Webtype ACL*, 523–524
- VTEP (Virtual Tunnel Endpoint) function**, 110–111
- VTI (Virtual-Tunnel Interfaces)**, 485
- VTY lines, AAA method lists**, 351–353
- vulnerabilities**
 - applications, 9
 - authentication-based vulnerabilities, 32–33
 - credential brute force attacks*, 33–34
 - cryptographic algorithms*, 33
 - default credentials*, 34
 - insecure direct object reference vulnerabilities*, 35
 - password cracking*, 33–34
 - rainbow tables*, 33–34
 - session hijacking*, 34
 - WEP*, 34
 - CVE, 9–10
 - CVSS, 67–71, 193, 595
 - defined, 9–10
 - hardware, 9
 - authentication-based vulnerabilities*, 32–35

- buffer overflows*, 39
- cookie manipulation attacks*, 37–38
- CVE, 30
- injection vulnerabilities*, 30–32
- NVD, 30
- OWASP Top 10 list, 40
- race conditions*, 38
- ret2libc attacks*, 39
- unprotected API*, 38–39
- XSRF, 37
- XSS, 35–37
- injection vulnerabilities, 30
 - command injections*, 32
 - HTML, 32
 - SQLi, 30–32
- misconfigurations, 9
- open source software, 40
- OS, 9
- SHA, 85–86
- shrinkwrap software, 9
- software vulnerabilities
 - authentication-based vulnerabilities*, 32–35
 - buffer overflows*, 39
 - cookie manipulation attacks*, 37–38
 - CVE, 30
 - injection vulnerabilities*, 30–32
 - NVD, 30
 - OWASP Top 10 list, 40
 - race conditions*, 38
 - ret2libc attacks*, 39
 - unprotected API*, 38–39
 - XSRF, 37
 - XSS, 35–37

- Tetration Vulnerability Dashboard, 595–596
- VDB updates, 458
- VXLAN (Virtual Extensible LAN), 110, 112–114, 120

W

- W3 schools, 135
- WADL (Web Application Description Language), 39, 137
- wallets (electronic/digital), Trojans, 20
- WannaCry, 23
- waterfall development methodology (cloud computing), 552–553
- watering holes, Trojans, 21
- WCCP (Web Cache Communication Protocol), 608
 - ASA configurations, 609–610, 612
 - IP spoofing, 615
 - switch configurations, 610–612
 - web traffic redirection to WSA, 609–610, 612
- weather-related threats, 12
- web authentication, 187–188
- WebEx, 167
- web forms, XSS testing, 37
- web proxy IP spoofing (WSA), 614–615
- WebGoat, 31
- WebSploit, 562
- Webtype ACL, 431, 523–524
- WEP, authentication-based vulnerabilities, 34
- white hack hackers, 14
- whitelists/blacklists (IP), 643–644
- Wi-Fi, IoT, 54
- WLAN (Wireless LAN), NetFlow deployment scenario, 244

worms, 16–17

WPAD (Web Proxy Auto-Discovery), 607

wrappers, malware distribution, 22

WSA (Web Security Appliance), 582, 604

- Content SMA, 624–628
- DNS, 607
- explicit forward mode, 606–608
- features of, 604–605
- policy configurations, 615–617
- proxies, 605–606
- reports, 617–619
- security services, 613–614
- SOCKS proxies, 607–608
- traffic redirection
 - policy-based routing*, 612–613
 - WCCP*, 609–610, 612
- transparent mode, 608–609
- WCCP, 608
 - ASA configurations*, 609–610, 612
 - IP spoofing*, 615
 - switch configurations*, 610–612
 - web traffic redirection to WSA*, 609–610, 612
- web proxy IP spoofing, 614–615
- WPAD, 607

WSDL (Web Services Description Language), 39, 137

X

X.500 standards, directory services, 97

XMPP and pxGrid, 183

XSD documents, 38

XSRF (Cross-Site Request Forgery), 37

XSS (Cross-Site Scripting), 32, 35–36

- cloud computing, 50
- DOM-based attacks, 36
- examples of, 36
- finding vulnerabilities, 36–37
- reflected XSS attacks, 36
- stored (persistent) XSS attacks, 36
- testing, 37

Y - Z

YANG models, 139–140

ZBFW (Zone-Based Firewalls), 411–412

zero-day exploits, 10

zero-trust, 161–167, 571

Zeus, 19

Zigbee, IoT, 53

zombies, 230

zone access, IPv4/IPv6, 373

Z-Wave, 53