EXAM CRAM

CCNP® and CCIE® Enterprise Core

ENCOR 350-401


Cram Sheet


Flash Cards


Practice Tests

DONALD BACHA

Table of Contents

Introduction . . . . . **xxiii**

Part I: Infrastructure

CHAPTER 1
Understanding Layer 2 . . . . . **1**

VLANs Overview 3
VLAN Assignment 4
802.1Q Trunking 7
Dynamic Trunking Protocol (DTP) 9
VLAN Trunking Protocol (VTP) 11
Inter-VLAN Routing 16
Spanning Tree Protocol Overview 19
Root Bridge, Root Port, and Designated Port Elections 20
Rapid Spanning Tree Protocol (RSTP) 25
Spanning Tree Protocol Tuning and Protection Mechanisms 28
Switch Priorities Overview 28
Multiple Spanning Tree Protocol (MST) 40
EtherChannels 47
Review Questions 57
Answers to Review Questions 58
Further Reading 58
What's Next? 58

CHAPTER 2
Understanding Layer 3: IGPs **59**

IP Routing Essentials 60
Routing Algorithms 61
Path Selection 62
Static Routing 65
Enhanced Interior Gateway Routing Protocol (EIGRP) 68
Neighbor Table 70
Topology Table 72
Routing Tables 75
EIGRP Authentication 76
EIGRP Named Mode 76
Route Summarization 78

Open Shortest Path First (OSPF) . . . 80
OSPF Cost . . . 81
OSPF Authentication . . . 82
OSPF Areas . . . 83
Neighbors and Adjacencies . . . 85
OSPF Packet Types . . . 87
Basic OSPF Configuration . . . 87
Router ID (RID) . . . 91
Passive Interfaces . . . 91
Default Route Advertisements . . . 91
OSPF Optimizations . . . 92
Link-State Advertisements (LSAs) . . . 92
OSPF Path Selection . . . 93
Route Summarization . . . 95
OSPFv3 . . . 95
Review Questions . . . 100
Answers to Review Questions . . . 101
Further Reading . . . 101
What's Next? . . . 101

CHAPTER 3
Understanding Layer 3: BGP . . . 103

BGP Fundamentals . . . 104
BGP Configuration and Verification . . . 112
Review Questions . . . 120
Answers to Review Questions . . . 120
Further Reading . . . 121
What's Next? . . . 121

CHAPTER 4
IP Services . . . 123

Network Time Protocol (NTP) . . . 124
Network Address Translation (NAT) . . . 134
Static NAT . . . 136
Dynamic NAT . . . 137
Port Address Translation (PAT) . . . 138
First-Hop Redundancy Protocols (FHRPs) . . . 143
Virtual Router Redundancy Protocol (VRRP) . . . 147
Gateway Load Balancing Protocol (GLBP) . . . 150
Object Tracking with FHRPs . . . 154

Multicast . . . 156
Multicast Fundamentals . . . 156
Multicast Group Addressing . . . 157
Internet Group Management Protocol (IGMP) . . . 157
Protocol Independent Multicast (PIM) . . . 161
Review Questions . . . 165
Answers to Review Questions . . . 165
Further Reading . . . 166
What's Next? . . . 166

CHAPTER 5
Enterprise Wireless . . . 167

Wireless Basics . . . 168
Radio Frequency (RF) . . . 168
Free Space Path Loss . . . 171
Received Signal Strength Indicator (RSSI) . . . 171
Signal-to-Noise Ratio (SNR) . . . 171
IEEE Wireless Standards . . . 172
Multiple Radios . . . 173
WLC and AP Operation and Pairing . . . 176
AP and WLC Interaction . . . 178
Wireless Roaming . . . 185
Troubleshooting WLAN Configuration and Client Connectivity Issues . . . 188
Review Questions . . . 191
Answers to Review Questions . . . 192
Further Reading . . . 192
What's Next? . . . 192

Part II: Security

CHAPTER 6
Device Access Control . . . 193

Cisco IOS CLI Session Overview . . . 194
Protection of Access to Cisco IOS EXEC Modes . . . 197
Secured Access with SSH . . . 203
Privilege Levels and Role-Based Access Control (RBAC) . . . 206
Authentication, Authorization, and Accounting (AAA) Overview . . . 210
TACACS+ Overview . . . 211
RADIUS Overview . . . 211
AAA Configuration for Network Devices . . . 212

Review Questions . . . 217
Answers to Review Questions . . . 217
Further Reading . . . 218
What's Next? . . . 218

CHAPTER 7
Infrastructure Security . . . 219

Access Control Lists (ACLs) Overview . . . 220
Types of ACLs . . . 224
Port ACLs (PACLs) and VLAN ACLs (VACLs) . . . 229
Control Plane Policing (CoPP) . . . 233
Review Questions . . . 236
Answers to Review Questions . . . 236
Further Reading . . . 237
What's Next? . . . 237

CHAPTER 8
Securing REST APIs . . . 239

REST API Security . . . 240
Review Questions . . . 245
Answers to Review Questions . . . 245
Further Reading . . . 245
What's Next? . . . 245

CHAPTER 9
Wireless Security . . . 247

Wireless Authentication Overview . . . 248
Open Authentication . . . 249
Pre-Shared Key (PSK) Authentication . . . 251
Extensible Authentication Protocol (EAP) Authentication . . . 254
WebAuth . . . 257
Review Questions . . . 262
Answers to Review Questions . . . 262
Further Reading . . . 262
What's Next? . . . 263

CHAPTER 10
Network Security Design . . . 265

Threat Defense . . . 266
Network Security Components . . . 270

TrustSec, MACsec . . . 279
TrustSec . . . 279
MACsec . . . 281
Review Questions . . . 284
Answers to Review Questions . . . 284
Further Reading . . . 285
What's Next? . . . 285

CHAPTER 11
Network Access Control . . . 287

Cisco Identity Services Engine (ISE) . . . 288
Network Access Control (NAC) . . . 290
Review Questions . . . 296
Answers to Review Questions . . . 296
Further Reading . . . 296
What's Next? . . . 297

Part III: Automation

CHAPTER 12
Anatomy of Python . . . 299

Interpreting Python Components and Scripts . . . 300
Python Overview . . . 300
Python Releases . . . 301
Setting Up Guest Shell . . . 301
Using Python . . . 302
Python Requirements . . . 309
Parsing Python Output to JSON . . . 310
Exception Handling . . . 311
Review Questions . . . 313
Answers to Review Questions . . . 313
Further Reading . . . 314
What's Next? . . . 314

CHAPTER 13
Building JSON Files . . . 315

Data Formats (XML and JSON) . . . 316
Extensible Markup Language (XML) . . . 317
JavaScript Object Notation (JSON) . . . 319
XML and JSON Comparison . . . 321

Review Questions . . . 323
Answers to Review Questions . . . 323
Further Reading . . . 324
What's Next? . . . 324

CHAPTER 14
YANG Data Modeling . . . 325

YANG Data Modeling . . . 326
Different YANG Models . . . 327
Review Questions . . . 332
Answers to Review Questions . . . 332
Further Reading . . . 332
What's Next? . . . 332

CHAPTER 15
DNA Center and vManage APIs . . . 333

APIs for Cisco DNA Center and vManage . . . 334
DNA Center API Integrations . . . 334
vManage API Integrations . . . 338
Review Questions . . . 344
Answers to Review Questions . . . 344
Further Reading . . . 344
What's Next? . . . 344

CHAPTER 16
Interpreting REST API Codes . . . 345

Interpreting REST API Response Codes . . . 346
HTTP Status Codes . . . 347
Review Questions . . . 349
Answers to Review Questions . . . 349
Further Reading . . . 349
What's Next? . . . 349

CHAPTER 17
EEM Applets . . . 351

Embedded Event Manager (EEM) . . . 352
EEM Architecture . . . 354
EEM Policies . . . 355
Review Questions . . . 362
Answers to Review Questions . . . 362

Further Reading . . . 362
What's Next? . . . 362

CHAPTER 18
Configuration Management and Orchestration . . . 363

Agent-Based Orchestration Tools . . . 365
Puppet . . . 365
Chef . . . 367
SaltStack . . . 369
Agentless Orchestration Tools . . . 372
Ansible . . . 372
Bolt . . . 375
Configuration Management and Orchestration Tools Comparison . . . 376
Review Questions . . . 378
Answers to Review Questions . . . 378
Further Reading . . . 378
What's Next? . . . 378

Part IV: Architecture

CHAPTER 19
Enterprise Network Design Principles . . . 379

Hierarchical LAN Design Model . . . 380
Access Layer . . . 381
Distribution Layer . . . 382
Core Layer . . . 382
Enterprise Network Architecture Options . . . 383
First-Hop Redundancy Protocols (FHRPs) . . . 392
Host Standby Router Protocol (HSRP) . . . 392
Virtual Router Redundancy Protocol (VRRP) . . . 396
Gateway Load Balancing Protocol (GLBP) . . . 397
Hardware Redundancy Mechanisms . . . 400
Stateful Switchover (SSO) . . . 400
Nonstop Forwarding (NSF) . . . 405
Review Questions . . . 407
Answers to Review Questions . . . 408
Further Reading . . . 408
What's Next? . . . 408

CHAPTER 20
Wireless LAN Deployments **409**

Wireless Deployment Models 410
Autonomous Wireless Deployments 411
Centralized Wireless Deployments 412
Cisco FlexConnect Wireless Deployments 415
Cloud-Based Wireless Deployments 418
Embedded Wireless Deployments 422
Wireless Location Services 427
Review Questions 430
Answers to Review Questions 431
Further Reading 431
What's Next? 431

CHAPTER 21
On-Premises vs. Cloud Infrastructure **433**

Cloud Infrastructure Basics 434
Cloud Services Models 438
Infrastructure as a Service (IaaS) 438
Platform as a Service (PaaS) 440
Software as a Service (SaaS) 441
Anything as a Service (XaaS) 442
Cloud Deployment Models 444
On-Premises or Cloud Infrastructure 447
Review Questions 449
Answers to Review Questions 449
Further Reading 450
What's Next? 450

CHAPTER 22
SD-WAN **451**

SD-WAN Overview 452
The Need for SD-WAN 453
Secure Automated WAN 454
Application Performance Optimization 455
Secure Direct Internet Access (DIA) 456
Multicloud 456
SD-WAN Architecture Components 459
vSmart Controllers 459
WAN Edge Routers 460

vBond Orchestrators . . . 461
vManage . . . 461
SD-WAN Considerations . . . 463
Review Questions . . . 465
Answers to Review Questions . . . 465
Further Reading . . . 466
What's Next? . . . 466

CHAPTER 23
SD-Access . . . **467**

SD-Access Overview . . . 468
SD-Access Architecture . . . 471
SD-Access Operational Planes . . . 474
SD-Access Fabric Roles and Components . . . 477
Control Plane Nodes . . . 478
Edge Nodes . . . 479
Intermediate Nodes . . . 480
Border Nodes . . . 480
Fabric Wireless LAN Controllers (WLCs) . . . 481
Fabric-Mode Access Points . . . 481
SD-Access Embedded Wireless . . . 481
Fabric in a Box . . . 482
Shared Services . . . 482
Review Questions . . . 484
Answers to Review Questions . . . 484
Further Reading . . . 484
What's Next? . . . 485

CHAPTER 24
QoS . . . **487**

The Need for QoS . . . 488
Packet Loss . . . 489
Delay . . . 490
Jitter . . . 491
Lack of Bandwidth . . . 491
QoS Models and Components . . . 493
Classification and Marking . . . 495
DSCPs and Per-Hop Behaviors (PHBs) . . . 497
Policing and Shaping . . . 497

Congestion Management and Congestion Avoidance . . . 499
Congestion Management (Queuing) . . . 499
Congestion Avoidance . . . 500
Wireless QoS . . . 500
Review Questions . . . 503
Answers to Review Questions . . . 503
Further Reading . . . 503
What's Next? . . . 504

CHAPTER 25
Switching . . . **505**

Traffic Forwarding Basics . . . 506
Forwarding Architectures . . . 511
Process Switching . . . 511
Fast Switching . . . 512
Cisco Express Forwarding (CEF) . . . 512
Tables Used in Switching . . . 515
Review Questions . . . 522
Answers to Review Questions . . . 522
Further Reading . . . 523
What's Next? . . . 523

Part V: Virtualization

CHAPTER 26
Basic Virtualization . . . **525**

Virtualization Overview . . . 526
Hypervisors . . . 527
Virtual Machines (VMs) . . . 532
Virtual Switching . . . 535
Network Virtualization . . . 537
Cisco Enterprise Network Function Virtualization (NFV) . . . 537
Cisco Enterprise NFV Architecture . . . 538
VNFs Supported in Cisco Enterprise NFV . . . 539
Cisco NFV Hardware Options . . . 539
Review Questions . . . 542
Answers to Review Questions . . . 543
Further Reading . . . 543
What's Next? . . . 543

CHAPTER 27
VRF Instances, GRE, and IPsec **545**

Virtual Routing and Forwarding (VRF) 546
VRF-Lite 547
Generic Routing Encapsulation (GRE) 552
IPsec VPNs 558
Site-to-Site VPNs 558
Dynamic Multipoint VPN (DMVPN) 559
Cisco IOS Virtual Tunnel Interfaces (VTIs) 560
Cisco IOS FlexVPN 561
IP Security (IPsec) 562
GRE Tunneling over IPsec 567
Review Questions 570
Answers to Review Questions 570
Further Reading 571
What's Next? 571

CHAPTER 28
Extending the Network Virtually **573**

Locator ID/Separation Protocol (LISP) 574
LISP Architecture 577
Virtual Extensible LAN (VXLAN) 580
Review Questions 585
Answers to Review Questions 585
Further Reading 586
What's Next? 586

Part VI: Network Assurance

CHAPTER 29
Troubleshooting **587**

Troubleshooting Overview 588
Using debug to Analyze Traffic 589
Troubleshooting with traceroute 593
Troubleshooting with ping 597
Simple Network Management Protocol (SNMP) 604
Review Questions 610
Answers to Review Questions 610
Further Reading 611
What's Next? 611

CHAPTER 30
Monitoring **613**

Syslog 614
NetFlow and Flexible NetFlow 620
Switch Port Analyzer (SPAN), Remote SPAN (RSPAN), and Encapsulated Remote SPAN (ERSPAN) 632
Remote SPAN (RSPAN) 634
Encapsulated Remote SPAN (ERSPAN) 635
Review Questions 639
Answers to Review Questions 640
Further Reading 640
What's Next? 640

CHAPTER 31
IP SLA and DNA Center **641**

IP SLA Overview 642
Cisco DNA Center Assurance 652
Review Questions 660
Answers to Review Questions 660
Further Reading 660
What's Next? 660

CHAPTER 32
NETCONF and RESTCONF **661**

NETCONF 662
RESTCONF 668
Review Questions 671
Answers to Review Questions 671
Further Reading 671
What's Next? 671

Glossary **673**

Index **695**

CHAPTER 6

Device Access Control

This chapter covers the following official ENCOR 350-401 exam objectives:

- 5.1 Configure and verify devices access control
- 5.1.a Lines and password protection
- 5.1.b Authentication and authorization using AAA

This chapter is divided into two sections. The first section looks at the configuration and verification of network device access control with usernames and passwords. It also covers the configuration and verification of role-based access control (RBAC) using privilege levels. The second section covers authentication, authorization, and accounting (AAA). It looks at the configuration and verification of network device access control on Cisco IOS devices using TACACS+ and RADIUS.

This chapter covers the following technology topics:

- Cisco IOS CLI Session Overview
  - Protection of Access to Cisco IOS EXEC Modes
  - Secured Access with SSH
  - Privilege Levels and Role-Based Access Control (RBAC)
- Authentication, Authorization, and Accounting (AAA) Overview
  - TACACS+ Overview
  - RADIUS Overview
  - AAA Configuration for Network Devices

CramSaver

If you can correctly answer these CramSaver questions, save time by skimming the Exam Alerts in this section and then completing the CramQuiz at the end of each section and the Review Questions at the end of the chapter. If you are in doubt at all, read everything in this chapter!

1. What are the first steps in securing user EXEC access to allow for secure network device access?
2. Which command option on remote CLI sessions is used to limit the session to use only a secure connection method?
3. What protocol does TACACS+ use for communication between a TACACS+ client (network device) and a TACACS+ server?
4. What are two of the high-level benefits of using a remote AAA server over local AAA services on each network device individually?

Answers

1. Configure passwords for local and remote CLI sessions.
2. **transport input ssh**
3. TCP port 49
4. Scalability and standardized authentication methods using RADIUS and TACACS+

Cisco IOS CLI Session Overview

Cisco IOS software provides several features that you can use to implement basic security for network devices' command-line sessions. These features include:

- Using different levels of authorization for CLI sessions to control access to commands that can modify the status of the networking device and for commands that are used to monitor the device
- Assigning passwords to CLI sessions
- Requiring users to log in to a networking device with a username
- Changing the privilege levels of commands to create new authorization levels for CLI sessions

You can establish IOS CLI sessions on Cisco IOS devices in two ways:

- **Local CLI sessions:** Local CLI sessions require direct access to the console port of the networking device. Local CLI sessions start in user EXEC mode. All of the tasks needed to configure and manage a networking device can be done using a local CLI session. The most common method for establishing a local CLI session is to connect a laptop to the console port of the networking device and then launch a terminal emulation application, like Putty, on the computer. The type of cable and connectors required and the settings for the terminal emulation application depend on the type of networking device that you are configuring. Some devices have an auxiliary (aux) port for remote administration through a dial-up modem. In most cases, this should be disabled with the **no exec** command under **line aux 0**.
- **Terminal lines and remote CLI sessions:** A remote CLI session is created between a host and a networking device by using a remote terminal access application, such as Telnet or SSH. Most of the tasks required to configure and manage a networking device can be done using a remote CLI session. The exceptions are tasks that interact directly with the console port (such as recovering from a corrupted operating system by uploading a new OS image over the console port) and interacting with the networking device when it is in ROMMON mode. SSH is a more secure alternative to Telnet. SSH provides encryption for the session traffic between the local management device and the networking device you are managing. Encrypting the session traffic with SSH prevents anyone who may have intercepted the traffic from decoding it.

With Cisco IOS networking devices, the word "lines" is used to refer to the software components that manage local and remote CLI sessions. You use the **line console 0** global configuration command to enter line configuration mode to configure options such as a password for the console port. Remote CLI sessions use lines that are referred to as vty lines. You use the **line vty** *line-number* [*ending-line-number*] global configuration command to enter line configuration mode to configure options such as a password for remote CLI sessions. Once you are in the line configuration mode, you can set the protocol you will be connecting over (for example, SSH).

Example 6.1 shows the console, auxiliary, and vty lines in the running configuration that are available on R1.

EXAMPLE 6.1 **Console, Auxiliary, and vty Lines in the Running Configuration**

```
R1#
R1# show running-config | section line
line con 0
line aux 0
line vty 0 4
R1#
```

Before we look at how to protect access to Cisco IOS EXEC modes, let's take a look at the five different types of passwords available in Cisco IOS:

- **Type 0 passwords:** Type 0 passwords are not encrypted and are stored in plaintext in the device configuration. The **enable password** command uses type 0 passwords. Type 0 passwords should not be used in a production environment.
- **Type 5 passwords:** Type 5 passwords use an MD5 hashing algorithm. These passwords are easily reversible with tools available on the Internet. The **enable secret** and **username** *username* **secret** commands use type 5 passwords.
- **Type 7 passwords:** Type 7 passwords uses the Vigenère cipher encryption algorithm, which is known to be weak. These passwords are easily reversible (in under 1 second) with tools available on the Internet. Type 7 password encryption is enabled with the **service password encryption** command.
- **Type 8 passwords:** Type 8 passwords use a Password-Based Key Derivation Function 2 (PBKDF2) with a SHA-256 hashed secret. Type 8 password security is considered good.
- **Type 9 passwords:** Type 9 passwords use the SCRYPT hashing algorithm. Type 9 passwords are considered the best passwords and should be used when supported.

Type 4 passwords were deprecated in IOS 15.3(3). The type 4 password hash was weaker than the type 5 (MD5) hash. Therefore, type 4 passwords should never be used. IOS 15.3(3) introduced support for type 8 and type 9 passwords, and these password types should always be used when supported.

Protection of Access to Cisco IOS EXEC Modes

This section looks at the steps you can take to secure both user and privileged EXEC modes.

The first step in creating secure network device access is to protect the user EXEC mode by configuring passwords for local and remote CLI sessions. You start by entering line configuration mode by selecting the line number for the console port (for example, **line console 0**). Once you are in that mode, you use the **password** command to assign a password to **line console 0**. You use the **login** command at **line console 0** to enable password checking at login.

Next, let's look at configuring a password for remote CLI sessions. After a password is configured for remote CLI sessions, the IOS device prompts for a password the next time you establish a remote CLI session with that device. Cisco IOS networking devices require that a password be configured for remote CLI sessions. If you attempt to start a remote CLI session with a device that does not have a password configured for remote CLI sessions, you get a message indicating that a password is required and that the password is not set. The remote CLI session will be terminated by the remote host.

To configure a password for remote CLI sessions, you start by entering the line configuration mode and selecting the vty line (for example, **line vty 0 4**). When you are in that mode, you use the **password** command as you do for the console line. You use the **login** command at the vty line to enable password checking at login.

Example 6.2 shows how to assign a password to the console, auxiliary, and vty lines and verify it in the running configuration.

EXAMPLE 6.2 **Configuring and Verifying Line Passwords**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line con 0
R1(config-line)# password Cisco123
R1(config-line)# login
R1(config-line)# line aux 0
R1(config-line)# password Cisco123
R1(config-line)# login
R1(config-line)# line vty 0 4
```

```
R1(config-line)# password Cisco123
R1(config-line)# login
R1(config-line)# end
R1#
R1# show running-config | section line
line con 0
 password Cisco123
 login
line aux 0
 password Cisco123
 login
line vty 0 4
 password Cisco123
 login
R1#
```

The previous section covers protection of access to both local and remote CLI sessions in user EXEC mode using line passwords. Now let's look at how to protect access to privileged EXEC mode. To add an additional layer of security, particularly for passwords that cross a network or that are stored with the configuration on a TFTP server, you can use the **enable secret** global configuration command.

Cisco recommends the use of the **enable secret** command over the **enable password** command because it uses an improved encryption algorithm. When you configure the **enable secret** command, it takes precedence over the **enable password** command. The two commands cannot be in effect simultaneously.

Let's look at the use of the **enable password** command to configure a password for privileged EXEC mode. The password you enter with the **enable password** command is stored as plaintext in the device's running configuration. You can encrypt the password for the **enable password** command in the configuration file of the networking device by using the **service password-encryption** command. However, the type 7 encryption level used by the **service password-encryption** command can be decrypted using tools available on the Internet, so it is not recommended for production deployments. The recommendation is to use the **enable secret** command because it provides strong encryption by hashing the password using type 5 passwords by default. However, on modern platforms, you can use type 8 or 9 passwords as well. You configure a password in privileged EXEC mode by using the command **enable secret** [**level** *level*] *unencrypted-password* | *encryption-type encrypted-password.* You can use the **show privilege** command to display the current level of privilege.

Example 6.3 shows the configuration and verification of protection of privileged EXEC mode using the **enable password** command. Note in the

verification that the password is stored in the running configuration in plaintext. This is because the default password, of type 0, was used. You can also set a type 7 password or set the EXEC level here. The command **service password-encryption** would make the password unreadable in the running configuration.

EXAMPLE 6.3 **Protecting Privileged EXEC with enable password**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# enable password ?
  0       Specifies an UNENCRYPTED password will follow
  7       Specifies a HIDDEN password will follow
  LINE    The UNENCRYPTED (cleartext) 'enable' password
  level   Set exec level password

R1(config)# enable password ExamCram123
 WARNING: Command has been added to the configuration using a type 0
password. However, type 0 passwords will soon be deprecated. Migrate
to a supported password type
R1(config)#
*Oct 28 23:00:00.922: %AAAA-4-CLI_DEPRECATED: WARNING: Command has
been added to the configuration using a type 0 password. However, type
0 passwords will soon be deprecated. Migrate to a supported password
type

R1(config)# do show run | include password
enable password ExamCram123
R1(config)#
R1(config)# service password-encryption
R1(config)# do show run | include password
enable password 7 106B11180834000A01557878
R1(config)# end
R1#
```


Example 6.4 shows the configuration and verification of protection of privileged EXEC mode using the **enable secret** command. This provides stronger encryption and is the recommended method to use. This example uses type 9 encryption. When using type 9, you need to type in the encrypted password or use the **algorithm-type** command to hash a plaintext **enable** secret. Note that the verification output shows the encrypted type 9 password.

EXAMPLE 6.4 **Protecting Privileged EXEC with enable secret**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
R1(config)# enable ?
  algorithm-type  Algorithm to use for hashing the plaintext 'enable'
secret
  password        Assign the privileged level password (MAX of 25
                  characters)
  secret          Assign the privileged level secret (MAX of 25
                  characters)

R1(config)# enable algorithm-type scrypt secret ?
  LINE   The UNENCRYPTED (cleartext) 'enable' secret
  level  Set exec level password

R1(config)# enable algorithm-type scrypt secret ExamCram123
R1(config)# do sho run | include secret
enable secret 9 $9$QlfhhreZrBM56f$VX4YG.yR/jHO/3gLFfTPqAw.
cdraNRDSKJoEOtCrC3Q
R1(config)# end
R1#
```

After you have protected access to user EXEC mode and privileged EXEC mode by configuring passwords for them, you can further increase the level of security on the device by creating usernames. You configure usernames to limit access to CLI sessions to a networking device to specific users. This is especially important if you are configuring a device to allow first-line technical support user access. These users typically would not need to run all commands available in privileged EXEC mode. For example, suppose you want technical support staff to be able to view the configuration on a device that will help them to troubleshoot network problems without being able to modify the configuration. In this case, you can create a username, configure it with privilege level 15, and configure it to run the **show running-config** command automatically. When a user logs in with the username, the running configuration will be displayed automatically.

There are three ways you can configure a username on a Cisco IOS device:

- Using the command **username** *username* **password** *password* configures a plaintext password (type 0).
- Using the command **username** *username* **secret** *password* provides type 5 encryption.
- Using the command **username** *username* **algorithm-type** [**md5** | **sha256** | **scrypt**] **secret** *password* provides type 5, type 8, or type 9 encryption, respectively.

The last option provides the highest level of security since it allows for the highest level of password encryption (type 8 or type 9). If the final option is not supported on a network device, then the second option should be used since it provides MD5 encryption. The first option should be avoided because it configures a plaintext password.

When you enable password authentication on a line by using the **password** command, you need to enable password checking. You do so by using the **login** command. This is what allows password use on the line. Once you have an alternate connection to the device, you can test the login. It is a good idea to have an alternate connection to a device if there is a problem logging in again using the line you made the changes on. The **login local** command allows for username/password pairs stored locally on the router to be used for the lines. By using the command **login local**, you can disable any password configured on lines.

To enable username and password authentication on a line, you need to do the following configuration:

- Create the user with the **username** command in global configuration mode, using one of the three options listed earlier in this section.
- Use the **login local** command in line configuration mode.

For remote CLI sessions, you can further protect the lines by using the **transport input** command. This command controls what protocols are allowed to access the vty lines. This can be configured with the command **transport input** {**all** | **none** | **telnet** | **ssh**}. The **all** option allows both Telnet and SSH access; **none** blocks Telnet and SSH; **telnet** allows only Telnet; and **ssh** allows only SSH access. Using **telnet ssh** allows both Telnet and SSH access. For the most secure access, the vty lines should be limited to SSH.

Example 6.5 shows the configuration and verification of usernames. The user **user1** is configured with a type 0 password, **admin1** is configured with a type 9 password, **tier1admin** is configured with a type 9 password (scrypt in this case), and **tier2admin** is configured with a type 8 password (sha256 in this case). The **login local** command is configured under the vty lines to tell it to use the router local user account database for authentication.

In this example, take note of the configured user accounts and the password types. **user1** with the type 0 password is shown in running configuration in plaintext. Privilege level 15 gives access to all commands, such as the **reload** command, and allows a user to make configuration changes on the device.

EXAMPLE 6.5 **Configuring Usernames and Passwords**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# username user1 password weakpassword
 WARNING: Command has been added to the configuration using a type 0
password. However, type 0 passwords will soon be deprecated. Migrate
to a supported password type
R1(config)# username admin1 privilege 15 secret admin1secret
R1(config)# username tier1admin algorithm-type scrypt secret
tier1adminsecret
R1(config)# username tier2admin algorithm-type sha256 secret
tier2adminsecret
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# end
R1#
R1# show running-config | include username
username user1 password 0 weakpassword
username admin1 privilege 15 secret 9 $9$iVS2wE3FxwTvDv$6k.
NoCSCi2af4T8HpWeO1lBaTUnJzelT8S6xEETp7AI
username tier1admin secret 9 $9$bIFEJkC8eW9Xyf$vXBZD.8ZSiHTcjpNVfuMWwX
vveegKfHCfNXgLZUYA9w
username tier2admin secret 8 $8$PLF4/9DTLkfoTf$820AEmeaZA2mNh1oNJjAYk6
bYKSlLhUn9pULnifodyo
R1#
```

Example 6.6 shows how to establish a Telnet session from R2 to R1 by using username-based authentication with the **tier1admin** username and type 9 password created earlier. You can see here that you can successfully connect and authenticate by using the **tier1admin** account.

EXAMPLE 6.6 **Verifying Username-Based Authentication for vty Lines**

```
R2#
R2# telnet 100.1.1.1
Trying 100.1.1.1 ... Open


User Access Verification

Username: tier1admin
Password:

! Password entered is not displayed by the router
R1>

R1#
```

```
R1# show line
  Tty Typ     Tx/Rx  A Modem  Roty AccO AccI  Uses  Noise Overruns Int
*   0 CTY            -    -      -    -    -    0      0    0/0      -
    1 AUX 9600/9600  -    -      -    -    -    0      0    0/0      -
*  578 VTY           -    -      -    -    -    2      0    0/0      -
   579 VTY           -    -      -    -    -    0      0    0/0      -
   580 VTY           -    -      -    -    -    0      0    0/0      -
   581 VTY           -    -      -    -    -    0      0    0/0      -
   582 VTY           -    -      -    -    -    0      0    0/0      -

Line(s) not in async mode -or- with no hardware support:
2-577
! the * in the output of the showline command indicates that the first
vty (0) is in use
! vty 0 is mapped to vty 578 automatically

R1#
```

ExamAlert

For the ENCOR exam, it is important to know the differences between the two SSH versions as well as the high-level steps for SSH configuration on Cisco devices.

Secured Access with SSH

SSH is a far more secure option than Telnet. Although Telnet is the most popular protocol used to access Cisco IOS devices, it is an insecure protocol. Its session packets are carried in plaintext, making it easy for someone to sniff and capture session information as it traverses the network. SSH provides encryption for session traffic between a device and a terminal access application. This prevents others from being able to intercept and decode the traffic.

SSH is available in two versions:

- **SSH Version 1 (SSHv1):** SSHv1 should be avoided because there are some flaws in its implementation, including its weak CRC-32 integrity check.
- **SSH Version 2 (SSHv2):** SSHv2 should be used when it is supported. The SSHv2 enhancement for RSA supports RSA-based public key authentication for a client and a network device. SSHv2 is not compatible with SSHv1.

Let us now take a look at the steps that are needed to set up a Cisco IOS device to run SSH:

1. Configure a hostname for the device, using the **hostname** *hostname* command.
2. Configure a domain name for the device, using the **ip domain-name** *domain-name* command.
3. Generate an RSA crypto key. Generating a key pair on the IOS device automatically enables SSH. When you generate an RSA key, you are prompted to enter a modulus length. A longer modulus length takes longer to generate, but it is more secure. You generate an RSA key with the **crypto key generate rsa** command.

Those three steps are mandatory. After you have taken those steps, you may need to set SSH to Version 2 because it is at SSHv1 by default on some platforms. You do this with the **ip ssh version 2** command. The other settings you can configure for the SSH service running on a device are the SSH timeout value and the authentication retries number. You do so with the command **ip ssh timeout** *seconds* **authentication-retries** *number*. Next, you set the transport input at the vty lines by using the **transport input ssh** command. Finally, also at the vty lines, you use the **login local** command to cause the local username and password on the router to be used for authentication.

For verification, you can use the **show ip ssh** command to view the version and configuration information for the SSH server. We can also use the **show ssh** command to show the status of the SSH server.

Example 6.7 demonstrates how to configure SSH, secure the vty lines to allow only SSH access, and verify connectivity from R2 to R1.

EXAMPLE 6.7 **Configuring and Verifying vty Access with SSH**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# username admin2 secret Cisco123
R1(config)# ip domain-name cisco.com
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 4096 for
your General Purpose Keys. Choosing a key modulus greater than 512 may
take a few minutes.
```

```
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)

R1(config)# ip ssh version 2
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# end
R1#

R2# ssh ?
  -c    Select encryption algorithm
  -l    Log in using this user name
  -m    Select HMAC algorithm
  -o    Specify options
  -p    Connect to this port
  -v    Specify SSH Protocol Version
  -vrf  Specify vrf name
  WORD  IP address or hostname of a remote system

R2# ssh -l admin2 -v 2 100.1.1.1


Password:
! Password entered is not displayed by the router

R1>
```


Finally, you can set a timeout for EXEC sessions that are left idle, which may pose a security risk. Under the line confirmation mode, you can use the **exec-timeout** *minutes seconds* command to set the timeout. The default setting is 10 minutes. Using **exec-timeout 0 0** and **no exec-timeout** disables the EXEC timeout. You should not use these commands this way in a production environment.

The **absolute-timeout** *minutes* command in the line configuration mode sets the interval for closing the EXEC session after a specified time has elapsed. This session is closed even if it is being used at the time of termination. You can use the **logout-warning** *seconds* command with the **absolute-timeout** command to notify users of an impending logout. By default, the user is given 20 seconds' notice before the session is terminated.

Example 6.8 shows how to configure EXEC and absolute timeouts and logout warning. For **line con 0**, a timeout value of 4 minutes is configured. For the vty lines, a value of 3 minutes and 30 seconds is configured. For the vty lines,

an absolute timeout of 10 minutes is configured, with a 120-second logout warning.

EXAMPLE 6.8 **Configuring EXEC and Absolute Timeouts**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# line con 0
R1(config-line)# exec-timeout 4 0
R1(config-line)# line vty 0 4
R1(config-line)# exec-timeout 3 30
!next we configure absolute timeout and logout warning
R1(config-line)# absolute-timeout 10
!logout warning is configured in seconds
R1(config-line)# logout-warning 120
R1(config-line)# end
R1#
```


Privilege Levels and Role-Based Access Control (RBAC)

Now that we have examined the various ways of securing user and privileged EXEC modes, let's take a look at the use of privilege levels and RBAC. By default, Cisco IOS devices have three privilege levels:

- **Privilege level 0:** Privilege level 0 allows for the use of five commands: **enable**, **disable**, **help**, **logout**, and **exit**.
- **Privilege level 1:** Privilege level 1 is the user EXEC mode that you saw configured earlier in this chapter, in the section "Protection of Access to Cisco IOS EXEC Modes." In this mode, it is not possible to make configuration changes.
- **Privilege level 15:** Privilege level 15 is the privileged EXEC mode you saw configured earlier in this chapter, in Example 6.5. (It is also configured in the next example.) In this mode, all of the IOS CLI commands are available.

The commands that you can run in user EXEC mode at privilege level 1 are a subset of the commands that you can run in privileged EXEC mode at privilege 15. You can configure additional privilege levels from 2 through 14 to provide customized access control. For example, you might want to allow a group of

technical support staff to configure only a specific set of interface-level commands on interfaces while preventing device-wide configuration privileges. You could configure this in global configuration mode by using the command **privilege** *mode* **level** *level* [*command string*]. After you create that technical support user and assign this privilege, the user will be allowed to enter the interface and execute the commands specified in the command string. You can verify the configuration with the **show privilege** command.

Example 6.9 shows how to set up privileges to allow a network operation staff member to do basic manipulation of an interface. This example shows how to create the user **user1noc** with a type 9 password and privilege level 5 configured. In this particular case, a user with the **user1noc** username will be allowed to shut, unshut, and assign an IP address on the interface because these are the only commands this configuration allows in privilege level 5 in interface configuration mode. A user who tries to type a command that is not allowed (such as the **description** command) gets the message "Invalid input detected."

EXAMPLE 6.9 **Configuring and Verifying a Username and a Privilege Level**

```
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# username user1noc privilege 5 algorithm-type scrypt secret
Cisco123
R1(config)# privilege exec level 5 configure terminal
R1(config)# privilege configure level 5 interface
R1(config)# privilege interface level 5 shutdown
R1(config)# privilege interface level 5 no shutdown
R1(config)# privilege interface level 5 ip address
R1(config)# end
R1#

R2# telnet 100.1.1.1
Trying 100.1.1.1 ... Open

User Access Verification

Username: user1noc
Password:

R1# show privilege
Current privilege level is 5
R1#
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```