This chapter includes the following topics:

- Cisco IOS Software security and configuration
- Catalyst 3550 security and configuration

CHAPTER 15

Basic Cisco IOS Software and Catalyst 3550 Series Security

Routers and Catalyst 3550 series switches are the predominant hardware components used in the CCIE Security lab exam. This chapter covers some of the basic security features that are available in Cisco IOS Software for routers and 3550 switches. The chapter is divided into two major parts:

- Cisco IOS Software security
- Catalyst 3550 security

The first part deals with the introduction and configuration of some of the basic router security features. The second part discusses configuration of basic security features on the 3550 switches. Although many more basic security features are available for routers and switches than those included in this chapter, here you concentrate on those features that are most likely to appear on the CCIE Security exam.

Cisco IOS Software Security

Routers are an important part of any network, and successful implementation of Cisco IOS Software security features increases router efficiency and, by association, the efficiency of the entire network. Cisco IOS Software includes a number of useful services. Unfortunately, many of them present a security concern. In this chapter, you find a short explanation of some of these services, their functionality, and how they can be misused by an attacker. Then you learn how to use the various Cisco IOS Software basic mechanisms that are designed to protect information.

Network Time Protocol Security

Network Time Protocol (NTP) is used for automatic time synchronization. Cisco networks use NTP to make timekeeping accurate and coordinated across the board. The use of NTP is highly recommended for security because having accurate time is important for intrusion and forensic analysis. NTP is typically deployed in a hierarchical fashion. All routers on the network should be made a part of the hierarchy, if possible. If an NTP hierarchy is not feasible, you should disable NTP. To prevent NTP from traversing the router altogether, apply an access list to an appropriate interface.

HTTP Server Security

To enable configuration and management of network devices remotely, Cisco IOS Software offers web-based Hypertext Transfer Protocol (HTTP) administration. Though the web-access features are quite common on Cisco routers, they facilitate not only a mechanism for monitoring and configuring but also for attacking a router. The HTTP traffic needs to be protected by securing the communication between the HTTP client and the HTTP server. Several security technologies are available for this task (HTTPS, SSL, SSH, and IPSec) which are discussed throughout this book. Of course, if web-based remote administration is not necessary, you should disable this feature.

Password Management

To control who can access the router command prompt, you can set various passwords for various access points to the router. You can configure the passwords for local console access or remote access via Telnet. This is done to prevent unauthorized changes to a router's behavior and also to protect information that can be learned by looking at the network statistics on a router. This chapter's password discussion concentrates on three types of passwords:

- Enable password
- Per-user passwords and privilege levels
- Line passwords

Enable Password

Enable password secures the privileged EXEC mode of a router. At this level, an administrator can view and change anything on the router. That is why such access needs to be closely guarded.

Privilege Levels

Additional controls are available in Cisco IOS Software to limit administrative access with various privilege levels. You can define different privilege levels for different passwords that permit a certain subset of commands to be configured by a user. Once the password is entered, the user is able to operate at the corresponding level. Cisco IOS Software supports a total of 16 privilege levels, ranging from 0 to 15. The default levels are 1 and 15. Level 1 is basic (or nonprivileged), and 15 is the privileged EXEC mode that was discussed in the preceding section.

Line Passwords

For remote administration, you can access Cisco routers via Telnet. Telnet occurs over virtual terminal lines (vty). Most Cisco IOS Software versions have five virtual terminals—0 through 4—

that support five simultaneous Telnet sessions. You should explicitly configure all the virtual terminals for security purposes. No password is configured on vty ports by default to deny all attempts to log in to a router remotely.

Access Lists

Cisco IOS Software uses *access lists*, also known as *Access Control Lists (ACLs)*, as security filters to permit or deny specific traffic from entering or exiting parts of the network. Access lists are used heavily on Cisco routers for restricting access to a router's services and for filtering traffic passing through the router. The router looks at each packet and determines whether to forward or drop the packet, based on the conditions that are specified in the access lists.

Access lists can include the source and destination addresses of the traffic, the protocol type, and so on. Access lists contain a list of statements that are arranged in sequential order that establishes the matching criteria. Each packet is checked against the list in the same order that the statements are positioned. When a match is found, the router processes the packet accordingly and does not go through the remainder of the statements. Therefore, you need to call out specific conditions before the more general ones. For more on access lists, refer to Chapter 16, "Access Control Lists."

Secure Shell

Secure Shell (SSH) service is a newer Cisco IOS Software feature that is intended for use in secure remote administration. To create a secure link between a client and a server, SSH uses Rivest, Shamir, and Adelman (RSA) public key cryptography. Therefore, the communication between the administrator's host and the router is encrypted. SSH is also used to prevent various kinds of network attacks. Currently, Cisco implements only version 1 of SSH, but remember to check for future updates.

NOTE The SSH client has been available since the Cisco IOS Software 12.1.3.T release.

Basic IOS Security Configuration

The following lessons and case studies are dedicated to basic Cisco IOS Software security configuration methods and are grouped into several scenarios, variations of which you are likely to encounter in the CCIE Security lab exam or in real life.

Lesson 15-1: Configuring Passwords, Privileges, and Logins

In this lesson, R8 is the router that needs to have basic Cisco IOS Software security features configured. Once R8 is configured, a remote host attempts to log in and perform some tasks.

This lesson covers the following configuration steps:

Step 1 Setting passwords

Step 2 Limiting connection time

Step 3 Configuring vtys and accessing the network remotely

Step 4 Creating user accounts

Step 5 Assigning privileges

Step 6 Local authentication, authorization, and accounting

Step 7 Remote administration with FTP

Step 8 Hiding Telnet addresses

Step 9 Verification

Step 1: Setting Passwords

First, you have to protect access to a router by setting various passwords. Prevent unauthorized login by configuring passwords on the console and virtual terminal lines. The syntax for both of them is identical, as follows:

```
R8(config-line)#password string
```

After the line passwords are set, you need to take care of the privileged EXEC level. You should not use the **enable password** command because it is not secure and can give away a system password. Instead, opt for the following command:

```
R8(config)#enable secret string
```

The **enable secret** command, as well as the username passwords described in "Creating User Accounts," later in this lesson, can be up to 25 characters long, including spaces, and are case sensitive. Example 15-1 demonstrates the application of passwords on R8. Note that both the console and the vty passwords appear scrambled. This is because **service password-encryption** is enabled on the router to hide the real string from a passerby.

Example 15-1 *Password Application on a Router*

```
R8#show run
version 12.2
service timestamps debug uptime
service timestamps log uptime
service password-encryption
```

continues

Example 15-1 *Password Application on a Router (Continued)*

```
!
hostname R8
!
enable secret 5 $1$uKVI$j1Y9WEzw7YIAWSkFwZZZB.
!
line vty 0 4
 password 7 1511021F0725
!
line con 0
 password 7 060506324F41
```

Step 2: Limiting Connection Time

For security reasons, you do not want to leave the connection to any port, be it console or remote connection, logged in indefinitely. If the connections are configured to time out automatically, the administrator is logged out by a router after a specified period if he forgets to do it himself. The syntax is the same for any line and is as follows:

```
R8(config-line)#exec-timeout minutes seconds
```

In Example 15-2, the console and auxiliary (aux) port are both configured to time out after a 5-minute interval.

Example 15-2 *Configuring a Timeout Period*

```
R8#show run
!
! Output omitted for brevity
!
line con 0
exec-timeout 5 0
password 7 05080F1C2243185E4B52
line aux 0
transport input all
exec-timeout 5 0
```

NOTE When you are in a lab-testing environment, a constant timeout can turn into a nuisance. If security is not an immediate concern, you can choose to set the timeout interval to infinity by using the **exec-timeout 0 0** command. However, you should never do so in real-world networking.

Step 3: Configuring vtys and Accessing the Network Remotely

As you know, vtys are used for remote network connections to the router. Generally, all the router's vtys have the same configuration. If there are extra vtys that are not used, it is a good practice to disable them with the **no line vty** command.

Applying an access list to vtys can effectively limit access to the router by specifying which connections are allowed. The command for assigning an access list to vtys is as follows:

```
R8(config-line)#access-class access-list in
```

Some of the protocols supported by the vtys (for example, rlogin and web) are not secure. To minimize the security risk, you can confine the acceptable type of connection to Telnet only with the following command:

```
R8(config-line)#transport input [telnet]
```

Example 15-3 shows IP access-list 5, which permits host 192.168.1.8. Applying access-list 5 to vty lines for inbound connections means that only one particular host can Telnet to R8.

Example 15-3 *The vty Configuration*

```
R8#show run
!
! Output omitted for brevity
!
access-list 5 permit 192.168.1.8
!
line vty 0 4
 access-class 5 in
 exec-timeout 5 0
 password 7 01302F377824
 transport input telnet
```

NOTE While configuring these commands, make sure that you are connected via an aux or console port. If you perform the commands while logged in to the router via Telnet, you might inadvertently disconnect yourself.

Step 4: Creating User Accounts

In this scenario, administrators log in according to the local router database. Each administrator receives his own username, password, and privilege level assigned, which indicates the level of control an administrator has over the router. The following command places a user in a local database:

```
R8(config)#username name privilege level password string
```

In Example 15-4, five administrators are assigned to the database. When they attempt to log in, they are authenticated by their username and corresponding password and are authorized to operate on the prescribed level.

Example 15-4 *Creating a Local Database*

```
R8#show run
!
! Output omitted for brevity
!
hostname R8
!
username admin privilege 3 password 7 02100A175809
username Sam privilege 15 password 7 05080F1C2243
username Jessie privilege 15 password 7 13061E010803
username Terry privilege 15 password 7 030752180500
username Joe privilege 5 password 7 01100F175804
```

Step 5: Assigning Privileges

Now that you have specified privilege levels for your users, you can assign a set of commands to a privilege level. Every user at the same privilege level can execute the same set. By default, every command in the Cisco IOS Software is designated for either level 1 or level 15. Level 0 exists, but it is rarely used. It includes following five commands:

- **disable**
- **enable**
- **exit**
- **help**
- **logout**

To change the default level and sign up certain commands to another level, use the following command:

```
R8(config)#privilege exec level level available-command
```

Keep in mind that for security reasons, you should move some commands that allow too much freedom for a lower level to a higher level, not the other way around. If you move higher-level commands, such as the **configure** command, down, you might enable a user to make unauthorized changes by letting him modify his own level to a higher one. Example 15-5 shows how privilege level 3 is limited to three commands:

- **telnet**
- **show ip route**
- **show startup**

Example 15-5 *Designating a Privilege Level*

```
R8(config)#privilege exec level 3 show start
R8(config)#privilege exec level 3 show ip route
R8(config)#privilege exec level 3 telnet
```

Step 6: Local Authentication, Authorization, and Accounting (AAA)

AAA technology is discussed in detail in Chapter 18, "AAA Services." Here, you are shown just a few AAA commands that make use of the local database that is configured in Steps 4 and 5 of this lesson. AAA has the following three separate functions:

- **Authentication**—Authentication identifies users before admitting them into a network.
- **Authorization**—Once a user is authenticated, authorization dictates what a user can accomplish on the network.
- **Accounting**—Accounting tracks the user's actions and logs them to monitor resource usage.

Example 15-6 illustrates the AAA commands configured on R8. To start an AAA process, the **aaa new-model** command is defined. The next command, **aaa authentication login default local**, names a local database as the one that is used for authentication on R8. The **aaa authorization config-commands** command enables AAA authorization of configuration commands specified by the **aaa authorization commands** statement that follows. The **aaa authorization exec default local** command specifies the local database as the source of authorization information, and the **aaa authorization commands 3 default local if-authenticated** command means that provided the user has been authenticated successfully, he is authorized by the router, after looking up the local database, to use the specified privilege level 3 commands. The latter command is helpful in the debugging process. Its practical usage is discussed in "Verification," later in this lesson.

Example 15-6 *AAA Configuration*

```
R8#show run
!
! Output omitted for brevity
!
hostname R8
!
aaa new-model
aaa authentication login default local
aaa authorization config-commands
aaa authorization exec default local
aaa authorization commands 3 default local if-authenticated
!
username admin privilege 3 password 7 02100A175809
```

NOTE User admin is authorized to operate at privilege level 3 only if the user accesses the router via vty. If the same user attempted to access R8 via console, the user would receive privilege level 15.

Step 7: Remote Administration with FTP

You can use File Transfer Protocol (FTP) to transfer configuration files to and from the router for remote administration. FTP is preferred because Trivial File Transfer Protocol (TFTP) does not support authentication and is, therefore, less secure and should not be used to transfer configuration files. The following commands are used to make the router FTP ready:

```
R8(config)#ip ftp source-interface interface-type number
R8(config)#ip ftp username name
R8(config)#ip ftp password string
```

The first command specifies the local interface that is set up for the FTP connection. The two subsequent commands create the username and password for authentication on the FTP server. Example 15-7 shows the FTP configuration on R8.

Example 15-7 *Configuring FTP*

```
R8#show run
!
! Output omitted for brevity
!
ip ftp source-interface FastEthernet0/0
ip ftp username anonymous
ip ftp password 7 1511021F0725
```

Step 8: Hiding Telnet Addresses

Normally, when you try to Telnet to a device, the router displays the address to which the connection is attempted along with other connection messages. This allows an unauthorized passerby to see it. To suppress the Telnet address, issue the following command:

```
R8(config)#service hide-telnet-address
```

Step 9: Verification

Example 15-8 demonstrates the output of the **debug aaa authentication** command followed by the **debug aaa authorization** command. The combination of these two commands shows the process a router goes through while authenticating and authorizing a user admin logging in from the remote host 192.168.1.6, permitted by access-list 5.

Example 15-8 *Debugging AAA*

```
R8#debug aaa authentication
R8#debug aaa authorization
Feb 28 17:48:46: AAA: parse name=tty66 idb type=-1 tty=-1
Feb 28 17:48:46: AAA: name=tty66 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=66
channel=0
Feb 28 17:48:46: AAA/MEMORY: create_user (0x8270E0D0) user='NULL' ruser='NULL' ds0=0
port='tty66' rem_addr
='130.100.26.2' authen_type=ASCII service=LOGIN priv=15 initial_task_id='0'
Feb 28 17:48:46: AAA/AUTHEN/START (1304267484): port='tty66' list='' action=LOGIN
service=LOGIN
Feb 28 17:48:46: AAA/AUTHEN/START (1304267484): using "default" list
Feb 28 17:48:46: AAA/AUTHEN/START (1304267484): Method=LOCAL
Feb 28 17:48:46: AAA/AUTHEN (1304267484): status = GETUSER
Feb 28 17:48:48: AAA/AUTHEN/CONT (1304267484): continue_login (user='(undef)')
Feb 28 17:48:48: AAA/AUTHEN (1304267484): status = GETUSER
Feb 28 17:48:48: AAA/AUTHEN/CONT (1304267484): Method=LOCAL
Feb 28 17:48:48: AAA/AUTHEN (1304267484): status = GETPASS
Feb 28 17:48:49: AAA/AUTHEN/CONT (1304267484): continue_login (user='admin')
Feb 28 17:48:49: AAA/AUTHEN (1304267484): status = GETPASS
Feb 28 17:48:49: AAA/AUTHEN/CONT (1304267484): Method=LOCAL
Feb 28 17:48:49: AAA/AUTHEN (1304267484): status = PASS
Feb 28 17:48:49: tty66 AAA/AUTHOR/EXEC (1491533337): Port='tty66' list='' service=EXEC
Feb 28 17:48:49: AAA/AUTHOR/EXEC: tty66 (1491533337) user='admin'
Feb 28 17:48:49: tty66 AAA/AUTHOR/EXEC (1491533337): send AV service=shell
Feb 28 17:48:49: tty66 AAA/AUTHOR/EXEC (1491533337): send AV cmd*
Feb 28 17:48:49: tty66 AAA/AUTHOR/EXEC (1491533337): found list "default"
Feb 28 17:48:49: tty66 AAA/AUTHOR/EXEC (1491533337): Method=LOCAL
Feb 28 17:48:49: AAA/AUTHOR (1491533337): Post authorization status = PASS_ADD
Feb 28 17:48:49: AAA/AUTHOR/EXEC: Processing AV service=shell
Feb 28 17:48:49: AAA/AUTHOR/EXEC: Processing AV cmd*
Feb 28 17:48:49: AAA/AUTHOR/EXEC: Processing AV priv-lvl=3
Feb 28 17:48:49: AAA/AUTHOR/EXEC: Authorization successful
```

Note that the **aaa authorization config-commands** commands and **aaa authorization commands 3 default local if-authenticated** commands of this scenario's AAA configuration were not yet set at the time the **debug** commands from Example 15-8 were issued. This resulted in the debug output not displaying the user's activity after the user has been authorized.

Example 15-9 shows the **debug** command output after **aaa authorization config-commands** commands and **aaa authorization commands 3 default local if-authenticated** commands have been applied. You can see that the user has issued the **show startup-config** command authorized for their privilege level.

Example 15-9 *Debugging AAA after the* **authorization config-commands** *Commands*

```
R8#show debug
General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on
```

Example 15-9 *Debugging AAA after the* **authorization config-commands** *Commands (Continued)*

```
Sep 28 17:40:45: AAA/AUTHEN (1358087791): status = GETUSER
Sep 28 17:40:45: AAA/AUTHEN/CONT (1358087791): Method=LOCAL
Sep 28 17:40:45: AAA/AUTHEN (1358087791): status = GETPASS
Sep 28 17:40:47: AAA/AUTHEN/CONT (1358087791): continue_login (user='admin')
Sep 28 17:40:47: AAA/AUTHEN (1358087791): status = GETPASS
Sep 28 17:40:47: AAA/AUTHEN/CONT (1358087791): Method=LOCAL
Sep 28 17:40:47: AAA/AUTHEN (1358087791): status = PASS
Sep 28 17:40:47: tty66 AAA/AUTHOR/EXEC (1731500233): Port='tty66' list='' service=EXEC
Sep 28 17:40:47: AAA/AUTHOR/EXEC: tty66 (1731500233) user='admin'
Sep 28 17:40:47: tty66 AAA/AUTHOR/EXEC (1731500233): send AV service=shell
Sep 28 17:40:47: tty66 AAA/AUTHOR/EXEC (1731500233): send AV cmd*
Sep 28 17:40:47: tty66 AAA/AUTHOR/EXEC (1731500233): found list "default"
Sep 28 17:40:47: tty66 AAA/AUTHOR/EXEC (1731500233): Method=LOCAL
Sep 28 17:40:47: AAA/AUTHOR (1731500233): Post authorization status = PASS_ADD
Sep 28 17:40:47: AAA/AUTHOR/EXEC: Processing AV service=shell
Sep 28 17:40:47: AAA/AUTHOR/EXEC: Processing AV cmd*
Sep 28 17:40:47: AAA/AUTHOR/EXEC: Processing AV priv-lvl=3
Sep 28 17:40:47: AAA/AUTHOR/EXEC: Authorization successful
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): Port='tty66' list='' service=CMD
Sep 28 17:40:55: AAA/AUTHOR/CMD: tty66 (1039984762) user='admin'
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): send AV service=shell
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): send AV cmd=show
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): send AV cmd-arg=startup-config
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): send AV cmd-arg=<cr>
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): found list "default"
Sep 28 17:40:55: tty66 AAA/AUTHOR/CMD (1039984762): Method=LOCAL
Sep 28 17:40:55: AAA/AUTHOR (1039984762): Post authorization status = PASS_ADD
```

Lesson 15-2: Disabling Services

Many services are offered by Cisco IOS Software. Although each service carries a useful function, it could present a potential security risk. When services are not used, you need to disable them. Otherwise, they open a security hole for an attacker to manipulate. This lesson is devoted to disabling unnecessary services on R8. Keep in mind that different Cisco IOS Software releases maintain different services on or off by default. If a service is off by default, disabling it does not appear in the running configuration. It is best, however, not to make any assumptions and to explicitly disable all unneeded services, even if you think they are already disabled.

The services covered in this lesson are as follows:

- Router name and DNS name resolution
- Cisco Discovery Protocol (CDP)
- TCP and UDP small servers
- Finger server

- NTP service
- BOOTP server
- Configuration auto-loading
- Proxy ARP
- IP source routing
- IP directed broadcast
- IP unreachables, redirects, and mask replies

Router Name and DNS Name Resolution

If no Domain Name System (DNS) server is specifically mentioned in the router configuration, by default all the name queries are sent to the broadcast address of 255.255.255.255. To alter the default behavior and turn off the automatic lookup, use the following command:

```
R8(config)#no ip domain-lookup
```

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a proprietary protocol that Cisco devices use to identify their directly connected neighbors. CDP is not frequently used and, like any other unnecessary local service, is considered potentially harmful to security. You can use the following commands to turn off CDP—globally and per interface:

```
R8(config)#no cdp run
R8(config-if)#no cdp enable
```

Disabling CDP per interface is a nice feature because it allows you to still run CDP for the parts of the network that need it.

TCP and UDP Small Servers

Another two services that you should also turn off are the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) small servers. They are included in the list of standard TCP and UDP services that hosts should provide but are seldom needed. Use the following commands to disable TCP and UDP small servers:

```
R8(config)#no service tcp-small-servers
R8(config)#no service udp-small-servers
```

Finger Server

Next, you need to make sure that the Cisco IOS Software support for the UNIX finger protocol is disabled. Having the finger service enabled allows a user to view other active users. There are

many known ways that the service can be misused and the information can fall into the wrong hands. To keep your network security in full force, you should consider turning off the finger service. After all, those who are not authorized to log in to the router have no business looking up those who do. Use the following command to disable the finger service:

```
R8(config)#no ip finger
```

NTP Service

If NTP, described earlier in "Network Time Protocol Security," is not used in the network, disable it with the following interface command:

```
R8(config-if)#ntp disable
```

BOOTP Server

In theory, BOOTP service might sound like a good idea. It is meant for use in networks where a centralized strategy of Cisco IOS Software deployment is implemented. One router can be used by other routers to load its operating system. However, the BOOTP protocol is seldom used, and it gives a hacker an opportunity to steal an IOS image. Therefore, in most situations, you should disable it using the following command:

```
R8(config)#no ip bootp server
```

Configuration Auto-Loading

The routers can find their startup configuration either in their own NVRAM or load it over the network. Obviously, loading in from elsewhere is taking a security risk. To disable the router's ability to get its configuration from the network, apply the following commands:

```
R8(config)#no boot network
R8(config)#no service config
```

Proxy ARP

Proxy Address Resolution Protocol (ARP) replies are sent to an ARP request destined for another device. When an intermediate Cisco device knows the MAC address of the destination device, it can act as a proxy. When an ARP request is destined for another Layer 3 network, a proxy ARP device extends a LAN perimeter by enabling transparent access between multiple LAN segments. This presents a security problem. An attacker can issue multiple ARP requests and use up the proxy ARP device's resources when it tries to respond to these requests in a denial-of-service (DoS) attack.

Proxy ARP is enabled on Cisco router interfaces. Disable it with the following interface command whenever it is not needed:

```
R8(config-if)#no ip proxy-arp
```

NOTE If, however, static routes use the interface as the destination instead of a next-hop router, proxy ARP is required.

IP Source Routing

An option is found in the header of every IP packet. The Cisco IOS Software examines the option and acts accordingly. Sometimes an option indicates source routing. This means that the packet is specifying its own route. Even though it is the default, this feature has several drawbacks. First, to allow source routing in the ISP environment means that a customer selects a route as they please. Also, this feature poses a known security risk, such as a hacker taking control of a packet's route and directing it through his network. So, if source routing is not necessary in your network, you should disable it on all routers by using the following command:

```
R8(config)#no ip source-route
```

IP-Directed Broadcast

If IP directed broadcast is enabled on a router's interface, it allows the interface to respond to the Internet Control Message Protocol (ICMP) requests directed to a broadcast address of its subnet. This can cause excessive traffic and possibly bring a network down, which is a tool often used by hackers in a smurf attack.

NOTE During a *smurf attack,* the ping requests sent to a broadcast address are forwarded to up to 255 hosts on a subnet. Because the return address of the ping request is spoofed to be the address of the attack target, all hosts that receive the ping requests reply to the attack target, flooding it with replies.

You can turn off IP directed broadcast capability on every interface with the following command:

```
R8(config-if)#no ip directed-broadcast
```

IP Unreachables, Redirects, and Mask Replies

ICMP messages that are automatically sent by Cisco routers in response to various actions can give away a lot of information, such as routes, paths, and network conditions, to an unautho-

rized individual. Attackers commonly use the following three types of ICMP message response features:

- **Unreachable**—A response to a nonbroadcast packet that uses an unknown protocol known as Protocol Unreachable, or a response to a packet that a responding device failed to deliver because there is no known route to a destination (Host Unreachable)
- **Redirect**—A response to a packet that notifies the sender of a better route to a destination
- **Mask Reply**—A response from a network device that knows a subnet mask for a particular subnet in an internetwork to a Mask Request message from a device that requires such knowledge

To disable the automatic messaging feature on interfaces, use the following commands:

```
R8(config-if)#no ip unreachables
R8(config-if)#no ip redirects
R8(config-if)#no ip mask-reply
```

Verification

Example 15-10 shows that all the services discussed in this lesson are disabled on R8. You do not see some of them in the running configuration output because of the default settings in this particular version of Cisco IOS Software.

Example 15-10 *Disabling Unnecessary Services*

```
R8#show run
version 12.2
hostname R8
!
! Output omitted for brevity
!
!
username admin privilege 3 password 7 02100A175809
username Sam privilege 15 password 7 05080F1C2243
username Jessie privilege 15 password 7 13061E010803
username Terry privilege 15 password 7 030752180500
username Joe privilege 5 password 7 01100F175804
no ip source-route
!
ip ftp source-interface FastEthernet0/0
ip ftp username anonymous
ip ftp password 7 1511021F0725
no ip domain-lookup
!
interface FastEthernet0/0
ntp disable
no cdp enable
!
interface FastEthernet0/1
```

continues

Example 15-10 *Disabling Unnecessary Services (Continued)*

```
ip address 192.168.1.1 255.255.255.0
no ip unreachables
no ip redirects
no ip mask-reply
no cdp enable
!
```

Lesson 15-3: Setting up a Secure HTTP Server

In this scenario, R8 needs to be configured as the HTTP server so that it allows remote management through the Cisco web browser interface. The syntax for the HTTP server command is as follows:

```
R8(config)#ip http server
```

Specifying the Port Number

You should change the HTTP port number from the default of 80 to something else to hide the HTTP server from an intruder. To modify the default, use the following command:

```
R8(config)#ip http port port-number
```

Specifying Authentication Technique

Next, you need to set up basic user authentication on your HTTP server. Although, you can use AAA services for this purpose, this example queries for the local database. The configuration of usernames and passwords in the database was discussed in the first lesson in "Configuring Passwords, Privileges, and Logins." Use the following command to set up basic user authentication on your local HTTP server:

```
R8(config)#ip http authentication [local]
```

Limiting Access to the Server

To limit access to the server, you can create an access list and then apply it to the HTTP configuration. To associate the list with the HTTP server access, generate the following command:

```
R(config)#ip http access-class access-list
```

Syslog Logging

You can choose to enable the logging of a router's events to a syslog server, including the HTTP-related activity. To specify syslog logging, use the following set of commands:

```
R8(config)#logging on
R8(config)#logging facility [syslog]
R8(config)#logging source-interface local-interface
R8(config)#logging syslog-server-address
R8(config)#logging trap [alerts]
```

The first command on the list, **logging on**, turns the logging on. The **logging facility** [**syslog**] command names a syslog server as the logging monitor. The **logging source-interface** *local-interface* command identifies local interface that forwards logs to the server. The **logging** *syslog-server-address* command points to the syslog server's IP address. The **logging trap** command sets up the trap level.

Verification

Example 15-11 displays the running configuration of R8. Notice the resolution of the HTTP commands. For example, the port number is changed to 8080. Access-list 11, permitting host 192.168.1.8, was created on R8. FastEthernet0/1 forwards logs to the server.

Example 15-11 *HTTP Configuration*

```
R8#show run
!
! Output omitted for brevity
!
ip http server
ip http port 8080
ip http access-class 1
ip http authentication local
access-list 11 permit 192.168.1.8
!
logging facility syslog
logging source-interface FastEthernet0/1
logging 192.168.1.7
logging trap alerts
```

Now that the HTTP server has been successfully configured, an authorized user can log in. Figures 15-1 and 15-2 show the browser login prompt and the postlogin screen, respectively.

Figure 15-1 *HTTP Login Prompt*

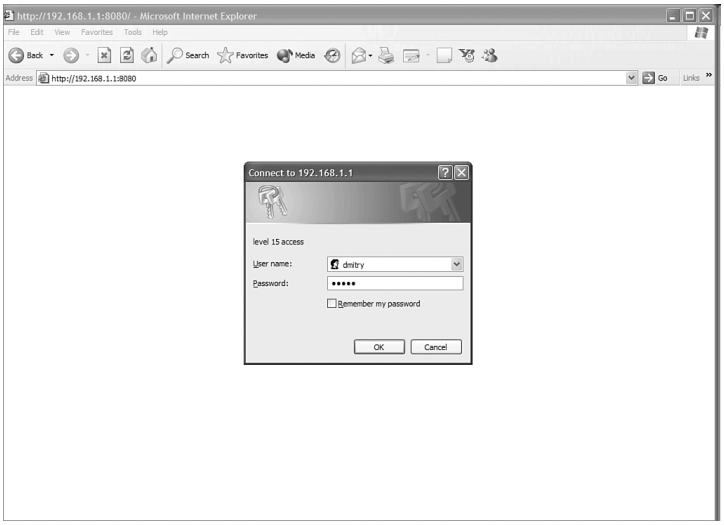
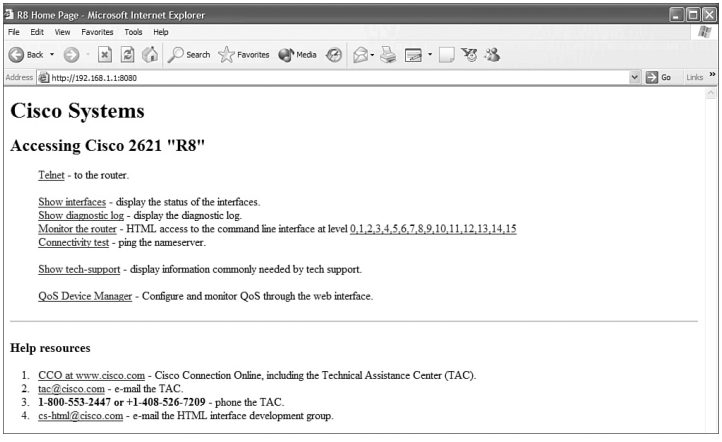


Figure 15-2 *Administrator's Browser Screen*

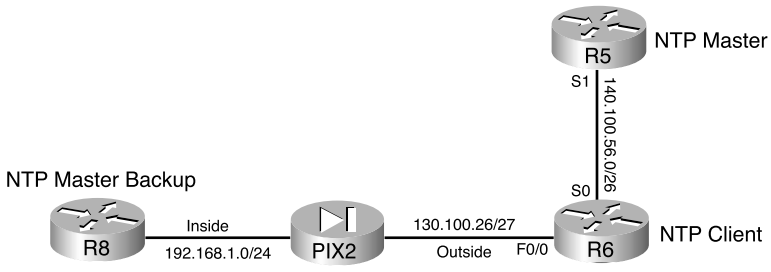


Case Study 15-1: Secure NTP Configuration

Figure 15-3 describes the network topology where R6 is a client of two NTP masters: R5 and R8. To throw in a twist, PIX2 is placed between R8 and R6. This case study is not meant as an

in-depth demonstration of the NTP protocol. The main goal is to achieve a functional, secure NTP configuration between the three routers using MD5 authentication.

Figure 15-3 *Network Topology for NTP Configuration*



This case study covers the following steps:

Step 1 Setting up time

Step 2 Setting up NTP relationships

Step 3 Configuring PIX2

Step 4 Restricting NTP access

Step 5 Configuring NTP authentication

Step 6 Verification

Step 1: Setting up Time

If you are using a local router as your time synchronization source, the first task you need to complete is to set the clock on the router that is to be your server, R5 in this case. The following command establishes the time (in military format) and date on the router:

```
R5>clock set hh:mm:ss day month year
```

Then, on all participating routers, set the time zone as compared to the Coordinated Universal Time (UTC). Also, configure the routers to automatically switch to daylight-saving time when appropriate. The following two commands identify the time zone and configure daylight-saving time for that zone:

```
R5(config)#clock timezone zone hours [minutes]
R5(config)#clock summer-time zone recurring [week day month hh:mm week day month hh:mm
  [offset]]
```