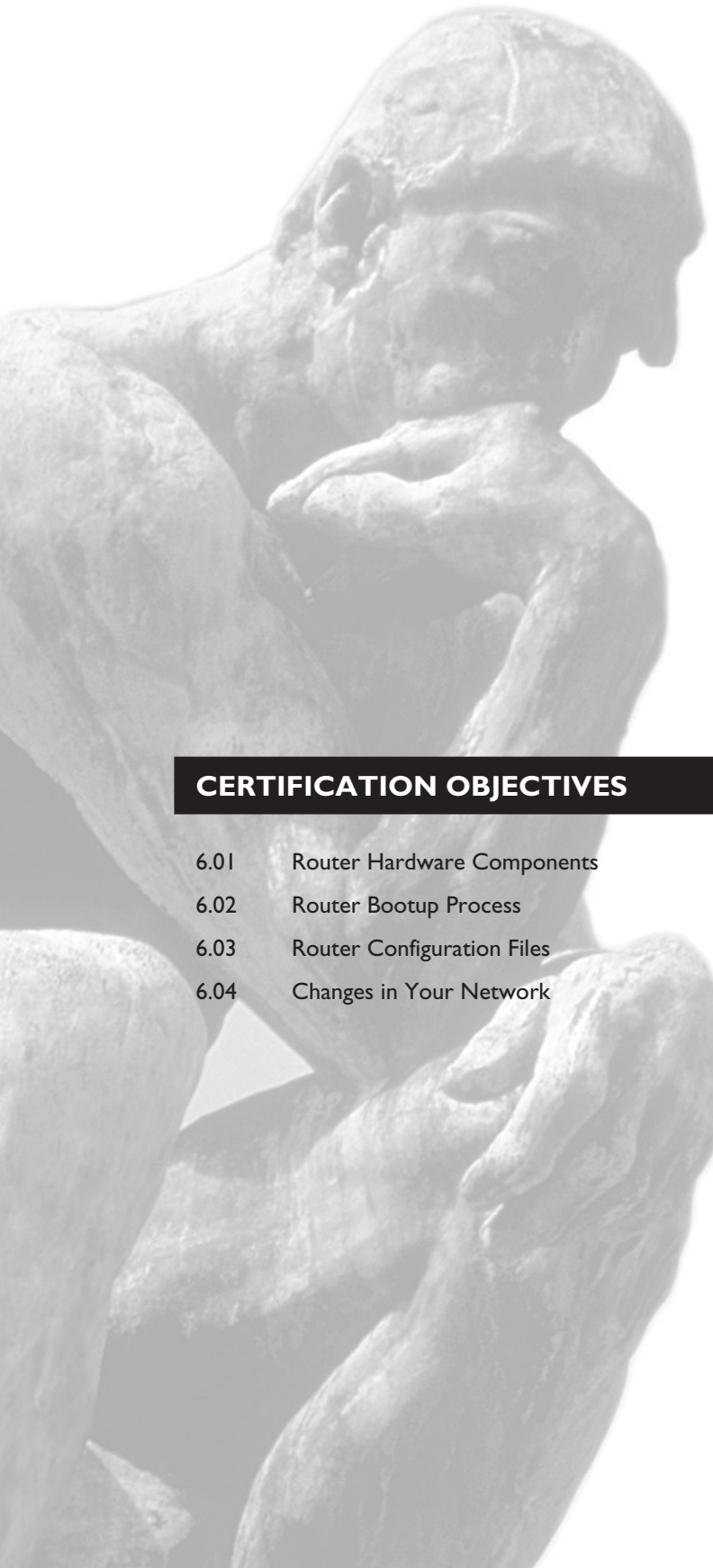


CCNA™
CISCO® CERTIFIED NETWORK ASSOCIATE

6

Managing Your Network Device

CERTIFICATION OBJECTIVES

This chapter covers important IOS features that you can use to manage your IOS device. Many of these features are supported across all IOS devices, but some of them are supported on only certain devices. This chapter focuses on these features as they relate to Cisco routers, beginning with how the router boots up, finds its operating system, and loads its configuration file, as well as how to back up and restore your IOS image. There are many tools that you can use on your router for troubleshooting connection problems, including the Cisco Discovery Protocol (CDP), ping, trace, telnet, and debug. These tools are discussed at the end of the chapter.

Router Hardware Components

Each IOS device has two main components: hardware and software. Almost every IOS-based router uses the same hardware and firmware components to assist during the bootup process, including the following: ROM (read-only memory), RAM (random access memory), flash, NVRAM (nonvolatile RAM), a configuration register, and physical interfaces. All of these components can affect how the router boots up, finds its operating system and loads it, and finds its configuration file and loads it. The following sections cover these components in more depth.

Read-Only Memory (ROM)

The software in ROM cannot be changed unless you actually swap out the ROM chip on your router. ROM is nonvolatile—when you turn off your device, the contents of ROM are not erased. ROM contains the necessary firmware to boot up your router and typically has the following four components:

- **POST (power-on self-test)** Performs tests on the router's hardware components.
- **Bootstrap program** Brings the router up and determines how the IOS image and configuration files will be found and loaded.
- **ROM Monitor (*ROMMON* mode)** A mini–operating system that allows you to perform low-level testing and troubleshooting, the password recovery procedure, for instance. To abort the router's normal bootup procedure of loading the IOS, use the CTRL-BREAK control sequence to enter *ROMMON* mode. The prompt in *ROMMON* mode is either ">" or "`rommon>`," depending on the router model.

- **Mini-IOS** A stripped-down version of the IOS that contains only IP code. This should be used in emergency situations where the IOS image in flash can't be found and you want to boot up your router and load in another IOS image. This stripped-down IOS is referred to as *RXBOOT* mode. If you see "`Router(rxboot)#`" in your prompt, then your router has booted up with the ROM IOS image. Not every router has a Mini-IOS image; on the other hand, some routers, such as the 7200, can store a full-blown IOS image here.

exam Watch

POST performs self-tests on the hardware. The bootstrap program brings the router up and finds the IOS image. ROMMON contains a mini-operating system used for low-level testing and debugging. The Mini-IOS is a stripped-down version of the IOS used for emergency booting of a router and is referred to as RXBOOT mode. All of these components are stored in ROM.

Other Components

Your router contains other components that are used during the bootup process, including RAM, flash, NVRAM, the configuration register, and the physical interfaces. The following paragraphs explain these components.

RAM is like the memory in your PC. On a router, it (in most cases) contains the running IOS image; the active configuration file; any tables (including routing, ARP, CDP neighbor, and other tables); and internal buffers for temporarily storing information, such as interface input and output buffers. The IOS is responsible for managing memory. When you turn off your router, everything in RAM is erased.

Flash is a form of nonvolatile memory in that when you turn the router off, the information stored in flash is not lost. Routers store their IOS image in flash, but other information can also be stored here. Note that some lower-end Cisco routers actually run the IOS directly from flash (not RAM). Flash is slower than RAM, a fact that can create performance issues.

NVRAM is like flash in that its contents are not erased when you turn off your router. It is slightly different, though, in that it uses a battery to maintain the information when the Cisco device is turned off. Routers use NVRAM to store their configuration files. In newer versions of the IOS, you can store more than one configuration file here.

The *configuration register* is a special register in the router that determines many of its bootup and running options, including how the router finds the IOS image and its configuration file. As you will see later in this chapter, you can manipulate this register to affect how your router boots up.

exam Watch

Flash is used to store the operating system and NVRAM is used to store the configuration file. The configuration register is used to determine how the router will boot up.

Every router has at least one port and one physical interface. *Ports* are typically used for management access; the console and auxiliary ports are examples. *Interfaces* are used to move traffic through the router; they can include media types such as Ethernet, Fast Ethernet, Token Ring, FDDI, serial, and others. These interfaces can be used during the bootup process—you can have the bootstrap program load the IOS from a remote TFTP server (instead of flash), assuming that you have a sufficient IP configuration on your router.

Router Bootup Process

A router typically goes through five steps when booting up:

1. The router loads and runs POST (located in ROM), testing its hardware components, including memory and interfaces.
2. The bootstrap program is loaded and executed.
3. The bootstrap program finds and loads an IOS image: Possible locations of the IOS image include flash, a TFTP server, or the Mini-IOS in ROM.
4. Once the IOS is loaded, the IOS attempts to find and load a configuration file, which is normally stored in NVRAM—if the IOS cannot find a configuration file, it starts up the *System Configuration Dialog* discussed in Chapter 5.
5. After the configuration is loaded, you are presented with the CLI interface (remember that the first mode you are placed into is *User EXEC* mode.

If you are connected to the console port, you'll see the following output as your router boots up:

```
System Bootstrap, Version 11.0(10c), SOFTWARE
Copyright (c) 1986-1996 by cisco Systems
2500 processor with 6144 Kbytes of main memory

F3: 5593060+79544+421160 at 0x3000060
```

```
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-I-L), Version 12.0(5)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Tue 15-Jun-99 19:49 by phanguye
Image text-base: 0x0302EC70, data-base: 0x00001000

<--output omitted-->

cisco 2501 (68030) processor (revision N) with
6144K/2048K bytes
of memory.
Processor board ID 18086269, with hardware revision
00000003
Bridging software.
X.25 software, Version 3.0.0.
Basic Rate ISDN software, Version 1.1.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

00:00:22: %LINK-3-UPDOWN: Interface Ethernet0, changed
state to up
00:00:22: %LINK-3-UPDOWN: Interface Serial0, changed
state to up
00:00:22: %LINK-3-UPDOWN: Interface Serial1, changed
state to up
00:00:23: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Ethernet0, changed state to up
00:03:13: %LINK-5-CHANGED: Interface Serial0, changed
state to administratively down
00:03:13: %LINK-5-CHANGED: Interface Serial1, changed
state to administratively down

Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-I-L), Version 12.0(5)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Tue 15-Jun-99 19:49 by phanguye

Press RETURN to get started!
```

There are a few things to point out here. First, notice that the router is loading the bootstrap program ("`System Bootstrap, Version 11.0(10c)`") and then the IOS image ("`IOS (tm) 2500 Software (C2500-I-L), Version 12.0(5)`"). During the bootup process, you cannot see the actual POST process. However, you will see information about the interfaces going up and/or down—this

exam Watch

When a router boots up, it runs POST, loads the bootstrap program, finds and loads the IOS, and loads its configuration file . . . in that order.

is where the IOS is loading the configuration and bringing up those interfaces that you previously activated. Sometimes, if the router has a lot of interfaces, the "`Press RETURN to get started!`" message is mixed in with the interface messages. Once the display stops, just hit ENTER to access *User EXEC* mode. This completes the bootup process of the router.

6.01. The CD contains a multimedia demonstration of booting up a Cisco router.

Bootstrap Program

As you saw in the bootup code example, the bootstrap program went out and found the IOS and loaded it. The bootstrap program goes through the following steps when trying to locate and load the IOS image:

1. Examine the configuration register value. This value is a set of four hexadecimal digits. The last digit affects the bootup process. If the last digit is between 0x2 and 0xF, then the router proceeds to the next step. Otherwise, the router uses the values shown in Table 6-1 to determine how it should proceed next.
2. Examine the configuration file in NVRAM for **`boot system`** commands, which tell the bootstrap program where to find the IOS. These commands are shown in the following paragraph.
3. If no **`boot system`** commands are found in the configuration file in NVRAM, use the first valid IOS image found in flash.
4. If there are no valid IOS images in flash, generate a TFTP local broadcast to locate a TFTP server (this is called a *netboot* and is not recommended because it is very slow and not very reliable for large IOS images).
5. If no TFTP server is found, load the Mini-IOS in ROM (*RXBOOT* mode).
6. If there is Mini-IOS in ROM, then the Mini-IOS is loaded and you are taken into *RXBOOT* mode; otherwise, the router either retries finding the IOS image or loads ROMMON and goes into *ROM Monitor* mode.

Table 6-1 contains the three common configuration register values in the fourth hex character of the configuration register that are used to influence the bootup

TABLE 6-1 Fourth Hex Character Configuration Register Values

Value in Last Digit	Bootup Process
0x0	Boot the router into *ROMMON* mode
0x1	Boot the router into *RXBOOT* mode using the Mini-IOS
0x2–0xF	Boot the router using the default boot sequence

exam Watch

The configuration register is used to influence how the IOS boots up.

process. The values in the configuration register are represented in *hexadecimal*, the register being 16 bits long.

For step 2 of the bootup process described in the last paragraph, here are the **boot system** commands that you can use to influence the order that the bootstrap program should use when trying to locate the IOS image:

```
Router(config)# boot system flash name_of_IOS_file_in_flash
Router(config)# boot system tftp IOS_image_name
IP_address_of_server
Router(config)# boot system rom
```

The **boot system flash** command tells the bootstrap program to load the specified IOS file in flash when booting up. Note that, by default, the bootstrap program loads the *first* valid IOS image in flash. This command tells the bootstrap program to load a different image. You might need this if you perform an upgrade and you have two IOS images in flash—the old one and new one. By default, the old one still loads first unless you override this behavior with the **boot system flash** command or delete the old IOS flash image.

You can also have the bootstrap program load the IOS from a TFTP server—this is not recommended for large images, since the image is downloaded via the UDP protocol, which is slow. And last, you can tell the bootstrap program to load the Mini-IOS in ROM with the **boot system rom command**. To remove any of these commands, just preface them with the **no** parameter.

on the job

The order that you enter the `boot system` commands is important, since the bootstrap program processes them in the order that you specify—once the program finds an IOS, it does not process any more `boot system` commands.

6.02. The CD contains a multimedia demonstration of using `boot system` commands on a router.

exam Watch

The `boot system` commands can be used to modify the default behavior of where the bootstrap program should load the IOS. When the bootstrap program loads, it examines the configuration file stored in NVRAM for `boot system` commands. If found, the IOS uses these commands to find the IOS. If no `boot system` commands are found, the router uses the default behavior in finding and loading the IOS image.

Configuration Register

As I mentioned in the last section, the configuration register is used by the bootstrap program to determine where the IOS image and configuration file should be loaded from. Once the router is booted up, you can view the configuration register value with the **`show version`** command:

```
Router> show version
Cisco Internetwork Operating System Software
IOS (tm) 3600 Software (C3640-JS-M), Version 12.0(3c),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1999 by cisco Systems, Inc.
Compiled Tue 13-Apr-99 07:39 by phanguye
Image text-base: 0x60008918, data-base: 0x60BDC000

ROM: System Bootstrap, Version 11.1(20)AA2,
EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)

Router uptime is 2 days, 11 hours, 40 minutes
System restarted by power-on
System image file is "flash:c3640-js-mz.120-3c.bin"

cisco 3640 (R4700) processor (revision 0x00) with 49152K/16384K
bytes of memory
<-- output omitted -->
125K bytes of non-volatile configuration memory.
32768K bytes of processor board System flash (Read/Write)

Configuration register is 0x2102
```

You need to go to the very bottom of the display in order to view the register value.

6.03. The CD contains a multimedia demonstration of using the `show version` command on a router.

Changing the Configuration Register from *Configuration* Mode

There are two ways of changing the configuration register value: from *Configuration* mode or from *ROMMON* mode. If you already have *EXEC* access to the router and want to change the register value, use this command:

```
Router(config)# config-register 0xhexadecimal_value
```

The register value is four hexadecimal digits, or 16 bits, in length. Each bit position in the register, though, indicates a function that the bootstrap program should take. Therefore, you should be very careful when configuring this value on your router.

The CD included with this book has a configuration register utility. Please take a look at this handy GUI-based tool from Boson--by selecting or deselecting specific boot options, the utility will *automatically* generate the correct register value for you.

When entering the register value, you must always precede it with "0x," indicating that this is a hexadecimal value. If you don't, the router assumes the value is decimal and *converts* it to hexadecimal. On a 2500 series router, the default configuration register value is 0x2102, which causes the router to use the default bootup process in finding and locating IOS images and configuration files. If you change this to 0x2142, this tells the bootstrap program that, upon the next reboot, it should locate the IOS using the default behavior, but *not* to load the configuration file in NVRAM; instead, you are taken directly into the *System Configuration Dialog*. This is the value that you will use to perform the password recovery procedure.

exam Watch

The default configuration register value is 0x2102, which causes a router to boot up using its default bootup process. You can see the configuration register value with the `show version` command. If you've changed this value, you will see the existing value and the value the router will use upon rebooting.

Changing the Configuration Register from ROM Monitor

Of course, one problem with the *Configuration* mode method of change the register value is that you must gain access to *Privilege EXEC* mode first. This can be a problem if you don't know what the passwords on the router are. There is a second method, though, that allows you to change the register value without having to log into the router. To

perform this method, you'll need console access to the router—you can't do this from the auxiliary port nor from a telnet session. Next, you'll turn the router off and then back on. As the router starts booting, you'll break into *ROMMON* mode with the router's break sequence. To break into the router, once you see the ROMMON program has loaded, you can, in most cases, use the CTRL-BREAK control sequence to break into *ROMMON* mode. Please note that this control sequence may be different, depending on the terminal program and operating system you are using on your PC.

Once in *ROMMON* mode, you can begin the process of changing the register value. There are two methods to do this, depending on the router that you have. Some of Cisco's routers, such as the 2600 and 3600, use the **confreg** script. This script asks you basic questions about the function and bootup process of the router. What's nice about the script is that you don't need to know the hexadecimal values for the configuration register, since the router will create it for you as you answer these questions. Here is an example of using this script:

```
rommon 5 > confreg
Configuration Summary
enabled are:
load rom after netboot fails
console baud: 9600
boot: image specified by the boot system commands
or default to: cisco2-C3600

do you wish to change the configuration? y/n  [n]:  y
enable  "diagnostic mode"? y/n  [n]:
enable  "use net in IP bcast address"? y/n  [n]:
disable "load rom after netboot fails"? y/n  [n]:
enable  "use all zero broadcast"? y/n  [n]:
enable  "break/abort has effect"? y/n  [n]:
enable  "ignore system config info"? y/n  [n]:
change console baud rate? y/n  [n]:
change the boot characteristics? y/n  [n]:

Configuration Summary
enabled are:
load rom after netboot fails
console baud: 9600
boot: image specified by the boot system commands
or default to: cisco2-C3600
do you wish to change the configuration? y/n  [n]:  n
rommon 6 >
```

Just as in the *System Configuration Dialog,* any information in brackets ("[]") represents default values. The first question that it asks is if you want to "change the configuration," which means change the register: answer "y" to continue. If you

answer "y" to "ignore system config info," the third hexadecimal digit becomes 4, making a 2500's register value appear as 0x2142. This option is used when you want to perform the password recovery procedure. The next-to-last question is "change the boot characteristics"—this question, if you answer "y," will repeat the questions again. Answer "n" to exit the script. If you make any changes, you are asked to save them ("do you wish to change the configuration?")—answer "y" to save your new register value.

on the job

As a shortcut, you could also execute the following command from ROMMON mode: confreg 0x2142.

Other routers, such as those in the 2500 series, do not support the **confreg** command. Instead, you'll need to use the following command to change the register value:

```
> o/r 0x4-digit_hexadecimal_value
```

In this situation, you must know the actual hexadecimal value that you will use in order to change the register value. You can use the **o** command to list the value in the register. Once you are done with either method, reboot the router. On many routers, just type in the letter "**i**" or "**b**" in *ROMMON* mode to boot it up.

6.04. The CD contains a multimedia demonstration of changing the configuration register in ROMMON mode and using the config-register command on a router.

exam Watch

When performing the password recovery procedure, break into ROMMON mode and change the configuration register value to 0x2142 and boot up the router. Once booted up, the router will ignore the configuration in NVRAM and take you into the System Configuration Dialog. Using CNTRL-C will break you out of this utility and take you to User EXEC mode. Enter Privilege EXEC mode and restore your configuration with the copy startup-config running-config command. The no shutdown command is not listed in the router's NVRAM configuration, so you will have to manually enable the interfaces. This is also true if you copy and paste a configuration into a router with its interfaces disabled, like a newly booted router.

Router Configuration Files

You've already had a basic introduction to configuration files in the last chapter. Remember that a configuration file contains the commands used to configure the router. Configuration files are typically located in one of three places: RAM, NVRAM, and/or a TFTP server. The configuration that the router is currently using is in RAM. You can back up, or save, this configuration to either NVRAM or a TFTP server.

As you may recall from the last chapter, the commands *related* to configuration files, even **show** commands, require you to be at *Privilege EXEC* mode. Also, only the 1900 switch automatically saves configuration files to NVRAM—you must manually do this on a router or 2950 switch. The following sections show you how to manipulate your configuration files on a router.

Saving Configuration Files

Chapter 5 explained how to save your configuration from RAM to NVRAM with the **copy running-config startup-config** command. When you execute this command, whatever filename (the default is "startup-config") you are copying to in NVRAM is completely overwritten. If you want to keep an old copy and a newer one in NVRAM, you'll need to specify a different name than "startup-config." Note that the **copy** command has two parameters. The first parameter refers to where the source information is (what you want to copy from), and the second parameter refers to where the destination is (what you want to copy to).

You can also back up your configuration to a TFTP server. This requires you to have TFTP server software on a server or PC and IP configured correctly on your router in order to access the server. The router command that you'll use is the **copy** command:

```
Router# copy running-config tftp
Address or name of remote host []? 192.168.1.11
Destination filename [Router-confg]?
!!
781 bytes copied in 5.8 secs (156 bytes/sec)
Router#
```

The syntax of the command for the 2950 switch is the same. You need to specify the IP address of the TFTP server as well as the filename that you want to save your configuration as. If the filename already exists on the server, the server *overwrites* the old file. After entering this information, you should see bang symbols ("!")

indicating the successful transfer of UDP segments to the TFTP server. If you see periods (“.”), this indicates an unsuccessful transfer. Plus, upon a successful transfer, you should also see how many bytes were copied to the server.

on the job

You can also back up configuration files to an FTP or RCP server. However, this is beyond the scope of this book.

The 1900 switch uses a different configuration file nomenclature for the **copy** commands. This is discussed in the section “Configuration File Nomenclature” later in this chapter.

You can also back up your saved configuration on your router or 2950 switch by replacing **running-config** in the preceding command with **startup-config**:

```
Router# copy startup-config tftp
```

This command backs up the configuration file in NVRAM to a TFTP server. As with the command before it, you will be prompted for the IP address of the TFTP server as well as the filename of the configuration file. Please note that if the file already exists on the TFTP server, the server will completely replace the old file with the new one.

6.05. The CD contains a multimedia demonstration of backing up the configuration file of a router.

Restoring Configuration Files

There may be situations when you have misconfigured your router or switch and wish to take a backed-up configuration file and load it back on to your Cisco device. You can do this by reversing the source and destination information in the **copy** command. There are actually three variations of the **copy** command that can accomplish this. Here is the first one:

```
Router# copy tftp startup-config
Address or name of remote host []? 192.168.101.1
Source filename []? router-confg
Destination filename [startup-config]?
Accessing tftp://192.168.101.1/p1r1-confg...
Loading Router-confg from 192.168.101.1 (via Ethernet0): !
[OK - 781/1024 bytes]
[OK]
781 bytes copied in 11.216 secs (71 bytes/sec)
Router#
```

In this example, the configuration file is copied from a TFTP server to NVRAM; if the file already exists there, it will be overwritten. Just as when backing up to a TFTP server, you must specify the server's IP address and the filename on the server.

You can also restore your configuration from a TFTP server to active memory:

```
Router# copy tftp running-config
```

There is one main different between moving the configuration from TFTP to NVRAM and moving it from TFTP to RAM. With the former method, the file in NVRAM is replaced with the one being copied; with the latter method, a *merge* process is used. During a merge process, the IOS updates commands that are common to both places—the new file and in RAM. The IOS also executes any new commands it finds in the uploaded configuration file. However, the IOS does not delete any commands in RAM that it does not find in the uploaded configuration file. In other words, this is *not* a replacement process. As an example, assume that you have a configuration file on a TFTP server that has IPX and IP information in it, but your RAM configuration has IP and AppleTalk. In this example, the router updates the IP configuration, adds the IPX commands, but leaves the AppleTalk commands as they are.

This process is also true if you want to restore your configuration from NVRAM to RAM with this command:

```
Router# copy startup-config running-config
```

6.06. The CD contains a multimedia demonstration of restoring the configuration file on a router.

Creating and Deleting Configuration Files

Besides backing up and restoring configuration files, you also need to know how to create and delete them. Actually, you already know how to create a basic configuration file by going into *Configuration* mode with the *Privilege EXEC* **configure** `terminal` command. When you are executing commands within this mode (whether by typing or pasting them in), the IOS is using a merge process (unless you use the **no** parameter for a command to delete or negate it).

You can also delete your configuration file in NVRAM by using the following command:

```
Router# erase startup-config
```

To verify the erasure, use the **show** `startup-config` command:

```
Router# show startup-config
%% Non-volatile configuration memory is not present
Router#
```

The 1900 switch is slightly different. The command to erase your configuration file is **delete nvram**. To view the configuration file, there is only one command: **show running-config**. Remember that the 1900 automatically saves its configuration to NVRAM. When you execute the **show running-config** command, you are actually looking at the active configuration, which *is* stored in NVRAM.

exam Watch

The copy command backs up and restores configuration files: copy running-config startup-config and copy running-config tftp back up the configuration file. copy startup-config running-config and copy tftp running-config or copy tftp startup-config restores the configuration file. The erase startup-config deletes the config file on a 2950 or a router, while the delete nvram deletes the config on a 1900.

6.07. The CD contains a multimedia demonstration of deleting the NVRAM configuration file of a router.

Configuration File Nomenclature

Starting with IOS 12.0 and later, Cisco introduced command and naming nomenclatures that follow IFS guidelines (what you are used to when you are entering a URL in a web browser address text box). Therefore, instead of entering a command and having a router prompt you for such additional information as the IP address of a TFTP server as well as the filename, you can now put all of this information on a single command line. Commands that reference configuration files and IOS images contain prefixes in front of the file type, which include the following:

- **bootflash** bootflash memory
- **flash** flash memory
- **flh** flash load helper log files

- **ftp** FTP server
- **nvram** NVRAM
- **rcp** Remote Copy Protocol (RCP) server
- **slot0** PCMCIA slot 0
- **slot1** PCMCIA slot 1
- **system** RAM
- **tftp** TFTP server

Let's take a look at an example. For instance, say that you want to back up your router's configuration from RAM to NVRAM. With the new syntax, you could type in the following:

```
Router# copy system:running-config nvram:startup-config
```

You don't always have to put in the type; for instance, in the preceding example, you could easily have entered this:

```
Router# copy running-config nvram:startup-config
```

To view the active configuration, you can use this command:

```
Router# more system:running-config
```

To delete all files in NVRAM, you can use this command:

```
Router# erase nvram:
```

To delete a specific file in NVRAM, you can use this form of the command:

```
Router# erase nvram:file_name
```

on the Job

The older style of entering configuration and IOS commands is still supported along with the new one.

The 1900, for the most part, uses the newer style of commands when dealing with the manipulation of configuration files. For instance, if you want to back up your configuration to a TFTP server, use the following syntax:

```
# copy nvram tftp://192.168.101.1/1900-config
```

In this example, the latter part of the command, referring to the TFTP server, follows the new nomenclature.

6.08. The CD contains a multimedia demonstration of using the new nomenclature for manipulating configuration files on a router.

6.09. The CD contains a multimedia demonstration of backing up and restoring configuration files on a 1900 switch.

Review of Configuration Files

It is important that you understand what action the IOS will take when it is either backing up or restoring a configuration file to a particular location. Table 6-2 summarizes this information for the routers.

exam Watch

Here is a quick way of remembering whether the IOS is using a merge or overwrite process. Anything copied into RAM uses a merge process, whereas any other copy operation is an overwrite process.

TABLE 6-2 Overview of IOS Process When Dealing with Configuration Files

Location (From)	Location (To)	Command	IOS Process
RAM	NVRAM	`copy running-config startup-config`	Overwrite
RAM	TFTP	`copy running-config tftp`	Overwrite
NVRAM	RAM	`copy startup-config running-config`	Merge
NVRAM	TFTP	`copy startup-config tftp`	Overwrite
TFTP	RAM	`copy tftp running-config`	Merge
TFTP	NVRAM	`copy tftp startup-config`	Overwrite
CLI	RAM	`configure terminal`	Merge

EXERCISE 6-1

Manipulating Your Router's Configuration Files

These last few sections dealt with the router's configuration files and how you manipulate them. This exercise will help you reinforce this material. You'll perform these steps on the 2600 router using Boson's NetSim™ simulator. You can find a picture of the network diagram for the simulator in the Introduction of this book. After starting up the simulator, click on the *LabNavigator* button. Next, double-click on *Exercise 6-1* and click on the *Load Lab* button. This will load the lab configuration based on Chapter 5's exercises. At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2600*.

1. Access the 2600 router's *Privilege EXEC* mode and save your router's active configuration to NVRAM. Verify the copy.

   Access *Privilege EXEC* mode: **`enable`**. Use the **`copy running-config startup-config`** command. Verify the copy: **`show startup-config`**.

2. Change the hostname on the router to *different* and then reload the saved configuration from the NVRAM into RAM. What is the hostname?

   Access *Configuration* mode (**`configure terminal`**) and use the **`hostname different`** command to change the router's name to *different*. Exit *Configuration* mode: **`end`**. Restore your configuration with **`copy startup-config running-config`**. Your prompt should change back to the previous name of the router (you might have to wait a few seconds for this to complete).

3. Erase your router's configuration in NVRAM. Examine the configuration file in NVRAM. Save the active configuration file to NVRAM. Examine the configuration file in NVRAM.

   Use the **`erase startup-config`** command to erase your configuration in NVRAM. Use the **`show startup-config`** command to verify the configuration file was deleted. Use the **`copy running-config startup-config`** command to save your configuration to NVRAM. Use the **`show startup-config`** command to verify that your router's configuration was backed up from RAM to NVRAM.

Now you should be more comfortable with the manipulating a router's configuration files. In the next section, you will learn how you should deal with changes in your network.

Changes in Your Network

When you decide to make any changes to your network, including the addition or deletion of devices, you should always do some preparation work *before* you make the change. Making changes can cause things to not function correctly, or not function at all, so you should always prepare beforehand. The following two sections cover the basics of handling changes.

Adding Devices

Before you add a device to your network, you should gather the following information and perform the following tasks:

1. Decide which IP address you'll assign to the device for management purposes.
2. Configure the ports of the device, including the console and VTY ports.
3. Set up your passwords for *User* and *Privilege EXEC* access.
4. Assign the appropriate IP addresses to the device's interface(s).
5. Create a basic configuration on the device so that it can perform its job.

Changing Devices

You will constantly be making configuration changes to your network to enhance performance and security. Before you make any changes to your network, you should *always* back up your configuration files. Likewise, before you perform a software upgrade on your Cisco device, you should always back up the old IOS image.

You should check a few things before loading the new image on your IOS device. First, does the new image contain all of the features that your previous image had? Or at least the features that you need? Also, does your router have enough flash *and* RAM to store and load the IOS image? You need to check these items out before proceeding to load the new image.

At times, you may need to upgrade the hardware or add a new module to your Cisco device. Some devices require you to turn them off to do the upgrade, while other devices do not. It is extremely important that you read the installation manual that comes with the hardware before performing the installation. If you install a hardware component into a device that requires the device to be turned off, and the device is running, you could damage your new component, or, worse, electrocute yourself.

Just remember that it is much easier to restore a backup copy than it is to recreate something from scratch. Whenever you make changes, always test the change to ensure that your Cisco device is performing as expected.

Router IOS Image Files

The default location of IOS images is in flash. Some routers have flash built into the motherboard, some use PCMCIA cards for storage, and some use a combination of both. At times, you will have to deal with the router's flash, when you want to do a router upgrade, for instance. To view your files in flash, use the **show flash** command:

```
Router# show flash
System flash directory:
File   Length    Name/status
1   10084696  c2500-js_l_120-3.bin
[10084696 bytes used, 6692456 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)
```

exam Watch

Use the show flash command or show version command to see how much flash memory is installed on your router.

In this example, you can see that flash holds one file 10,084,696 bytes in length that is called *c2500-js_l_120-3.bin*. Below this, you can see how much flash is used (about 10MB), how much is available (about 6MB), and the total amount of flash on the router (16MB). You can also see how much flash you have installed on your router with the **show version** command.

6.10. The CD contains a multimedia demonstration of viewing flash on a router.

Naming Conventions for IOS Images

Cisco has implemented a naming convention for its IOS images, allowing you to tell the platform, software version, and features included in the image . . . just by looking at the name of the image file. As example, I'll use the image name from the previous **show flash** command: "c2500-js_l-120-3.bin."

The "c2500" refers to the name of the platform that the image will run on. This is important because different router models have different processors; and an image compiled for one processor or router model will typically *not* run on a different model. Therefore, it is very important that you load the appropriate image on your router.

The "js" refers to the features included in this IOS version, commonly referred to as the *feature set*. In this example, "j" refers to the enterprise edition, while "s" means that the IOS image has enhanced features.

The "l" (the letter "l," not the number "1") indicates where the IOS image is run from. The "l" indicates relocatable and that the image can be run from RAM. Remember that some images can run directly from flash, depending on the router model. If you see "mz" or "z," this means that the image is compressed and must be uncompressed before running.

The "120-3" indicates the software version number of the IOS. In this instance, the version is 12.0(3). And the ".bin" at the end indicates that this is a binary image.

on the job

The naming nomenclature discussed here applies to IOS images that are either included on your IOS device when you buy it from Cisco or applied when you download them from Cisco's web site. However, the name, in and of itself, has no bearing on the actual operation of the IOS when it is loaded on your IOS device. For instance, you can download an image from Cisco and rename it to "poorperformance.bin," and this will have no impact on the IOS device's performance.

exam Watch

Cisco uses a specialized naming convention when naming their IOS images. This convention contains the platform image, the feature set, whether or not the image is relocatable or is compressed, and the IOS version and revision numbers.

Before Upgrading the IOS Upgrade

This section and the next section discuss how to upgrade and backup the IOS software on your router. Before you upgrade the IOS on your router, you should first back up the existing image to a TFTP server. There are two reasons that you might want to do this. First, your flash might not be large enough to support two images—the old one and the new. If you load the new one and you experience problems with it, you'll probably want to load the old image back onto your router. Second, Cisco doesn't keep every software version available on their web site. Older versions of the IOS are hard to locate, so if you are upgrading from an old version of the IOS, I would highly recommend backing it up first.

Before you back up your IOS image to a TFTP server, you should also perform the following checks:

- Is the TFTP server reachable (test with the **ping** command)?
- Is there enough disk space on the TFTP server to hold the IOS image?
- Does the TFTP server support the file nomenclature that you want to use?
- Does the file have to exist on the TFTP server before you can perform the copy? (This is true with certain TFTP servers in the Unix world.)

Once you have performed these checks, you are ready to continue with the backup process.

Backing Up an IOS Image

To back up your IOS image, you'll use the **copy flash tftp** command. When you execute this command, you'll be prompted for the following information:

- The name of the IOS image in flash to back up—use the **show flash** command to get this name
- The TFTP server's IP address
- The name that you want to call the image when it is copied to the TFTP server

Here is an example of the use of this command:

```
Router# copy flash tftp
Source filename []? c3640-js-mz.120-11
Address or name of remote host []? 192.168.1.1
Destination filename [c3640-js-mz.120-11]?
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<--output omitted-->
6754416 bytes copied in 64.452 secs (105537 bytes/sec)
Router#
```

As the image is backed up, you should see a bunch of exclamation points ("!") filling up your screen—this indicates the successful copy of a UDP segment. If you see periods (".") instead, this indicates a failure. After a successful copy operation, you should see the number of bytes copied as well as how long it took. Compare the number of bytes copied to the file length in flash to verify that the copy was really successful.

6.11. The CD contains a multimedia demonstration of backing up the IOS flash image on a router.

Loading an IOS Image

If you want to upgrade your IOS or load a previously saved IOS image, you'll need to place the IOS image on a TFTP server and use the **copy tftp flash** command. You'll be prompted for the same information as you were when you used the **copy flash tftp** command; however, the process that takes place after you enter your information is different.

After your enter your information, the IOS first verifies that the image exists on the TFTP server. If the file exists on the server, the IOS then prompts you if you want to erase flash. Answer "y" if you don't have enough space in flash for the older image(s) as well as the new one. If you answer "y," flash is erased and reprogrammed; as this step proceeds, you will see a list of "**e**"s appear on the screen.

After flash is initialized, your router pulls the IOS image from the TFTP server. Just as in the copy operations with configuration files, a bunch of "!"s indicate successful copies, while "."s indicate unsuccessful copies.

on the job

Not every IOS version has the same upgrade process, so what you see on your router may be different from this book, especially if you are running IOS versions 11.x or earlier.

Here is example of loading an IOS image into your router:

```
Router# copy tftp flash
Address or name of remote host []? 192.168.1.1
Source filename []? c3640-js-mz.120-7
Destination filename [c3640-js-mz.120-7]?
%Warning:There is a file already existing with this name
Do you want to over write? [confirm] y
Accessing tftp://192.168.1.1/c3640-js-mz.120-7...
Erase flash: before copying? [confirm] y
Erasing the flash filesystem will remove all files! Continue? [confirm] y
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Loading c3640-js-mz.120-7 from 192.168.1.1 (via FastEthernet0/0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<--output omitted-->
[OK - 6754416/13508608 bytes]

Verifying checksum...  OK (0xCAF2)
6754416 bytes copied in 66.968 secs (102339 bytes/sec)
Router#
```

In this example, the router noticed that the name of the image that exists on the TFTP server is the same one that is in flash and verifies that you want to overwrite

exam Watch

Use the `copy flash tftp` command to back up the IOS image and the `copy tftp flash` command to restore or upgrade the IOS. The `reload` command reboots the router.

it. After the router copies the IOS image to flash, you must reboot your router in order for it to use the new image. There are two ways you can reboot your router: turn it off and back on or use the *Privilege EXEC* **`reload`** command. The first method is a hard reboot, and the second one is a soft reboot.

If you place an incorrect image on your router, for instance, a 3600 series image on a 2500 series router, the router will not reboot. You'll need to break into *ROMMON* mode and either do a TFTP boot or boot from the Mini-IOS in ROM.

on the Job

The 2950 uses the same process as a Cisco router for backing up and loading IOS images. The 1900 switch doesn't support backing up of images; however, you can load an IOS image with this command:
`copy tftp://IP_address/IOS_image opcode.`

6.12. The CD contains a multimedia demonstration of loading an IOS flash image on a router.

IOS Troubleshooting

The remainder of this chapter focuses on troubleshooting tools that you can use on your routers and switches. One of your first troubleshooting tasks is to figure out in which layer of the OSI Reference Model things are not working. By narrowing the problem down to a specific layer, you've greatly reduced the amount of time that you'll need in order to fix it. Cisco has a wide variety of tools that you can use. Here is a list of the more common tools and what layer of the OSI Reference Model that they can be used for in troubleshooting:

- **`show interfaces`** command Layer-2 (covered in Chapter 5)
- Cisco Discovery Protocol (CDP) Layer-2
- **`ping`** command Layer-3
- **`traceroute`** command Layer-3
- **`telnet`** command Layer-7
- **`debug`** commands Layers 2–7

The following sections covers these tools in more depth.

Cisco Discovery Protocol (CDP)

CDP is a Cisco proprietary data link layer protocol that was made available in version 10.3 of the router IOS. Almost every Cisco device supports CDP, including Cisco routers and Catalyst switches. For those devices that support CDP, CDP is *enabled* by default. CDP messages received from one Cisco device, by default, are not forwarded to any other devices behind it. In other words, you can see CDP information only about other Cisco devices *directly* connected to you.

exam Watch

The important point to make about CDP is that if you are receiving CDP frames from a Cisco neighbor, then at least the data link layer is functioning correctly.

CDP Information

CDP, as mentioned in the last paragraph, works at the data link layer. However, since CDP uses a SNAP frame type, not every data link layer media type is supported. The media types are supported: Ethernet, Token Ring, FDDI, PPP, HDLC, ATM, and Frame Relay.

The information shared in a CDP packet about a Cisco device includes the following:

- Name of the device configured with the **`hostname`** command
- IOS software version
- Hardware capabilities, such as routing, switching, and/or bridging
- Hardware platform, such as 2600, 2950, or 1900
- The layer-3 address(es) of the device
- The interface the CDP update was generated on

exam Watch

CDP messages are generated every 60 seconds as multicast messages on each of its active interfaces. Interfaces supported include ATM, Ethernet, FDDI, Frame Relay, HDLC, and PPP. CDP information is not propagated to other Cisco devices behind your directly connected neighboring Cisco devices.

CDP Configuration

As I mentioned in the last section, CDP is enabled on all Cisco CDP-capable devices when you receive your product from Cisco. On Cisco routers and the 2950 switch, you can globally disable or enable it with this command:

```
Router(config)# [no] cdp run
```

There is no command on the 1900 switch to globally disable it. However, on Cisco routers, the 2950, and the 1900, you can enable or disable CDP on an interface-by-interface basis:

```
Router(config)# interface type [slot_#/]port_#
Router(config-if)# [no] cdp enable
```

Since CDP doesn't use up much resources (a small frame is generated once a minute), it is recommended to keep it enabled unless your router is connected to the Internet; and then you should at least disable CDP on the external, or public, interface. At a minimum, the information is only 80 bytes in length. There are other, optional commands related to CDP, such as changing the update and hold down timers, but these commands are beyond the scope of this book.

6.13. The CD contains a multimedia demonstration of disabling and enabling CDP on a router.

CDP Verification

To see the status of CDP on your Cisco device, use this command:

```
Router# show cdp
Global CDP information:
Sending CDP packets every 60 seconds
Sending a holdtime value of 180 seconds
Sending CDPv2 advertisements is  enabled
```

As you can see from this output, CDP is enabled and generating updates every 60 seconds. The hold down timer is 180 seconds. This timer determines how old a CDP neighbor's information is kept in the local CDP table without seeing a CDP update from that neighbor.

You can also see the CDP configuration on an interface-by-interface basis by adding the **interface** parameter to the **show cdp** command:

```
Router# show cdp interface
Serial0 is up, line protocol is up, encapsulation is HDLC
Sending CDP packets every 60 seconds
```

```
Holdtime is 180 seconds
Ethernet0 is up, line protocol is up, encapsulation is ARPA
Sending CDP packets every 60 seconds
Holdtime is 180 seconds
```

To see a summarized list of the CDP neighbors that your Cisco device is connected to, use the **show cdp neighbors** command:

```
Router# show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge,
B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater

Device ID    Local Intrfce Holdtme Capability Platform Port ID
0090AB56EEC0 Fas 0/1        170      T S      1900     B
```

In this example, there is one device connected to a router with a device ID of "0090AB56EEC0," which is a 1900 switch without a hostname. In this situation, the MAC address of the switch is used as the ID. This update was received on `fastethernet 0/1` on this device 10 seconds ago (hold-down timer of 170 seconds). The `Port ID` refers to the port at the remote side—`B` is `fastethernet 0/27` on a 1900 switch—that the device advertised the CDP message from.

You can add the optional **detail** parameter to the preceding command to see the details concerning the connected Cisco device. On Cisco routers and the 2950, you can also use the **show cdp entry *** command. Here is an example of a CDP detailed listing:

```
Router# show cdp neighbor detail
-------------------------
Device ID: RouterA
Entry address(es):
IP address: 172.16.1.1
IPX address: 10.0000.0000.1111
Platform: cisco 4500,  Capabilities: Router
Interface: Ethernet0/0,  Port ID (outgoing port): Ethernet0/1
Holdtime : 137 sec

Version :
Cisco Internetwork Operating System Software
IOS (tm) 4500 Software (C4500-J-M), Version 11.1.10,
MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Mon 07-Apr-97 19:51 by dschwart

<-- output truncated -->
```

In this example, you can see that the connected device is a 4500 series router running IOS 11.1(1) and has IP and IPX addresses configured on the connected interface.

For the router and 2950, you can list the details of a specific neighbor with this command:

```
Router# show cdp entry neighbor's_name
```

The advantage of this approach over that in the preceding example is that this command lists only the specified neighbor. This command is not supported on the 1900.

The 2950 and router support one additional command, which allows you to view CDP traffic statistics:

```
Router# show cdp traffic
Total packets output: 350, Input: 223
Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
No memory: 0, Invalid: 0, Fragmented: 0
```

If you are receiving CDP traffic, then the data link layer is functioning correctly.

6.14. The CD contains a multimedia demonstration of viewing CDP information on a router.

exam Watch

CDP is enabled, by default, on all Cisco devices. CDP updates are generated as multicasts every 60 seconds with a hold-down period of 180 seconds for a missing neighbor. The `no cdp run` command globally disables CDP, while the `no cdp enable` command disables CDP on an interface. Use `show cdp neighbors` to list out your directly connected Cisco neighboring devices. Adding the `detail` parameter will display the layer-3 addressing configured on the neighbor.

Layer-3 Connectivity Testing

As you saw in the preceding section, CDP can be very useful in determining if the data link layer is working correctly. You can even see the layer-3 address(es) configured on your neighboring device and use this for testing layer-3 connectivity. Besides using CDP, you could also use the **`show interfaces`** command for data link layer testing.

However, the main limitation of these two tools is that they don't test layer-3 problems. Cisco, nevertheless, has tools for testing layer-3 connectivity. This chapter

exam Watch

The simple `ping` and `traceroute` commands can be executed from either User or Privilege EXEC modes. However, the extended ping and traceroute commands can only be executed from Privilege EXEC mode.

focuses on two of these commands: **`ping`** and **`traceroute`**. Both of these commands come in two versions: one for *User EXEC* mode and one for *Privilege EXEC*. The *Privilege EXEC* version provides additional options and parameters that can assist you in your troubleshooting process. Both of these tools are supported on Cisco routers and the 2950 switches; however, only the *User EXEC* mode of **`ping`** is supported on the 1900 switches. The following sections cover these tools in more depth.

Using Ping

Ping (Packet Internet Groper) was originally developed for the IP protocol stack to test layer-3 connectivity. With IP, the ICMP protocol is used to implement ping. However, Cisco's IOS has expanded the **`ping`** command to support other protocols, including: Apollo, Appletalk, CLNS, DECnet, IP, IPX, Vines, and XNS. Cisco uses ping to test layer-3 connectivity with other protocols in a (typically) proprietary fashion.

on the Job

When troubleshooting PC problems, first determine if the user can ping their loopback address: `ping 127.0.0.1`. If this fails, then there is something wrong with the TCP/IP protocol stack installation on the PC. Next, have the user try to ping their configured IP address. If this fails, there is something wrong with their IP address configuration. Next, have the user ping the default gateway. If this fails, then there is either something wrong with the configured default gateway address, the default gateway itself, or the subnet mask value configured on the user's PC.

exam Watch

Ping uses ICMP echo messages to initiate the test. If the destination is reachable, the destination responds back with an echo reply message for each echo sent by the source. Both the `ping` and `traceroute` commands test layer-3 connectivity.

Simple `ping` Command To execute a simple ping from either *User* mode or *Privilege EXEC* mode, enter the **`ping`** command on the CLI and follow it with the IP address or hostname of the destination:

```
Router> ping destination_IP_address_or_host_name
```

Here is a simple example of using this command:

```
Router> ping 192.168.101.1
Type escape sequnce to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1,
timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
round-trip min/avg/max = 2/4/6 ms.
Router>
```

In this example, five test packets were sent to the destination and the destination responded back to all five, as is shown by the exclamation marks ("**`!`**"). The default timeout to receive a response from the destination is two seconds—if a response is not received from the destination for a packet within this time period, a period ("**`.`**") is displayed.

There are basically two reasons you might see a "**`.`**" in the output—a response was received, but after the timeout period, or no response was seen at all. If a response was received, but after the timeout problem, this might be because an ARP had to take place to learn the MAC address of a connected device or because of congestion. Here are two examples: "**`.!!!!`**" and "**`!!..!`**". If devices have to perform ARPs to get the MAC address of the next device, then you'll typically see the first example in your output. However, if your output looks like the second example, you're probably experiencing congestion or performance problems. Table 6-3 shows examples of ping messages that you might see in your output. The bottom of the output shows the success rate—how many replies were received and the minimum, average, and maximum round-trip times for the ping packets sent (in milliseconds). This information can be used to detect if there is a delay between you and the destination.

Extended `ping` Command The IOS routers and 2950 switches support an extended **`ping`** command, which can only be executed at *Privilege EXEC* mode. To execute this command, just type **`ping`** by itself on the command line:

```
Router# ping
Protocol [ip]:
Target IP address: 192.168.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]
Extended commands [n]: y
Source address:
Type of service [0]:
Set DF bit in IP header? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Number of hops [9]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1,
timeout is 2 seconds:
<-- output truncated -->
```

TABLE 6-3 Output Codes for the **ping** Commands

Ping Output	Explanation
.	A response was not received before the timeout period expired.
!	A response was received within the timeout period.
U	A remote router responded that the destination is unreachable—the segment is reachable, but not the host.
N	A remote router responded that the network is unreachable—the network cannot be found in the routing table.
P	A remote device responded that the protocol is not supported.
Q	Source quench, telling the source to slow its output.
M	The ping packet needed to be fragmented, but a remote router couldn't perform fragmentation.
A	The ping packet was filtered by a device (administratively prohibited).
?	The ping packet type is not understood by a remote device.
&	The ping exceeded the maximum number of hops supported by the routing protocol.

Here is an explanation of the parameters that might be asked of you when you execute this command:

- **Protocol** The protocol to use for the ping (defaults to IP)
- **Target IP address** The IP address or hostname of the destination to test
- **Repeat count** How many echo requests should be generated for the test (defaults to 5)
- **Datagram size** The size, in bytes, of the ping packet (defaults to 100)
- **Timeout in seconds** The amount of time to wait before indicating a timeout for the echo (defaults to two seconds)
- **Extended commands** Whether or not the remaining questions should also be asked (this defaults to "no")
- **Source address** The IP address that should appear as the source address in the IP header (this defaults to the IP address of the interface the ping will use to exit the IOS device)
- **Type of service** The IP level for QoS (defaults to 0)
- **Set DF bit in the IP header?** Whether or not the ping can be fragmented when it reaches a segment that supports a smaller MTU size (the default is no—don't set this bit). Sometimes a misconfigured MTU can cause performance problems. You can use this parameter to pinpoint the problem, since a device with a smaller MTU size will not be able to handle the large packet.
- **Data pattern** The data pattern that is placed in the ping. It is a hexadecimal four-digit (16-bit) number (defaults to 0xABCD) and is used to solve cable problems and crosstalk on cables.
- **Loose, Strict, Record, Timestamp, Verbose** IP header options (defaults to "none" of these). The *record* parameter records the route that the ping took—this is somewhat similar to traceroute. If you choose *record*, you will be asked for the maximum number of hops that are allowed to be recorded by the ping (defaults to 9, and can range 1–9).
- **Sweep range of sizes** Send pings that vary in size. This is helpful when trying to troubleshoot a problem related to a segment that has a small MTU size (and you don't know what that number is). This defaults to *n* for no.

To break out of a `ping` or `traceroute` command, use the* CTRL-SHIFT-6 *break sequence.

6.15. The CD contains a multimedia demonstration of using the simple and extended `ping` commands on a router.

Using Traceroute

One limitation of ping is that this command will not tell you where, between you and the destination, layer-3 connectivity is broken. Traceroute, on the other hand, will list each router along the way, including the final destination. Therefore, if there is a layer-3 connection problem, with traceroute, you'll know at least where the problem begins. Like the **`ping`** command, **`traceroute`** has two versions: one for *User EXEC* mode and one for *Privilege EXEC*. The following two sections cover the two different versions.

Simple `traceroute` Command The simple **`traceroute`** command, which works at both *User* and *Privilege EXEC* modes, has the following syntax:

```
Router> trace destination_IP_address_or_host_name
```

Here is an example of this command:

```
Router> traceroute www.trace.com
Type escape sequence to abort.
Tracing the route to www.monkeys.com (200.1.90.6)
1 router.dealgroup.com (192.168.1.1) 732 msec 8 msec 7 msec
2 router11.myisp.COM (200.1.89.1) 9 msec 8 msec 8 msec
3 www.trace.com (200.1.90.6) 10 msec 11 msec 11 msec
```

In this example, the destination was three hops away—each hop is listed on a separate line. For each destination, three tests are performed, where the round-trip time is displayed for each test. If you don't see round-trip time, this indicates a possible problem. Table 6-4 shows other values that you might see instead of the round-trip time.

In certain cases, for a specific destination, you might see three asterisks ("*****") in the output—you shouldn't be alarmed if you see this, since a variety of things can cause it; for instance, there may be an inconsistency in how the source and destination devices have implemented traceroute, or the destination may be configured not to reply to these messages. However, if you continually find the same destination repeated in the output with these reply messages, this indicates a layer-3 problem starting with either this device or the device before it.

TABLE 6-4 Traceroute Messages

TracerouteOutput	Explanation
*	Either the wait timer expired while waiting for a response or the device did not respond at all.
A	The trace packet was filtered by a remote device (administratively prohibited).
U	The port of the device is unreachable (the destination received the trace packet but discarded it).
H	The destination is unreachable (the destination segment was reachable, but not the host).
I	The user interrupted the traceroute process.
N	The network is unreachable (the destination segment was not reachable).
P	The protocol is unreachable (the device doesn't support traceroute).
Q	Source quench.
T	The trace packet exceeded the configured timeout value.
?	The device couldn't identity the specific trace type in the trace packet.

If you have DNS lookups enabled on your Cisco device (on routers, this is the `ip domain-lookup` command), the IOS will attempt to resolve the IP address to a domain name before printing the output line for that device. If your traces seem to take a long time, this is usually the culprit. You can disable DNS lookups on your router with the `no ip domain-lookup` command.

Extended `traceroute` Command The extended **`traceroute`** command is similar to the extended **`ping`** command and requires *Privilege EXEC* mode access to execute it:

```
Router# trace
Protocol [ip]:
Target IP adddress: "IP address of the destination"
Source address:
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
```

```
Port number [33434]:
Loose, Strict, Record, Timestamp, Verbose [none]:
<-- output truncated -->
```

Some of these options are the same used by ping. Here is an explanation of the other options. The "`Numeric Display`" option turns off a DNS lookup for the names of the routers and the destination. The "`Time to Live`" options specify how many hops the trace is allowed to take. The "`Loose`" option tells the router that the hops you specify must appear in the trace path, but other routers can appear as well. The "`Strict`" option restricts the trace path to only those routers that you specify. The "`Record`" option specifies the number of hops to leave room for in the trace packet, and the "`Timestamp`" option allows you to specify the amount of space to leave room for in the trace packet for timing information. The "`Verbose`" option is automatically selected whenever you choose any of the options from this question; it prints the entire contents of the trace packet.

exam Watch

The `ping` command uses ICMP to test layer-3 connectivity to a device. The `traceroute` command lists each router along the way to the destination and is typically used to troubleshoot routing problems.

One important item to point out about this command is that if there is more than one path to reach the destination, the **`traceroute`** command will test *each* path.

6.16. The CD contains a multimedia demonstration of using the simple and extended `traceroute` commands on a router.

Layer 7 Connectivity Testing

The **`ping`** and **`traceroute`** commands can test only layer-3 connectivity. If you can reach a destination with either of these two commands, this indicates that layer-3 and below is functioning correctly. You can use other tools, such as telnet, to test the application layer. If you can telnet to a destination, then all seven layers of the OSI Reference model are functioning correctly. As an example, if you can telnet to a machine but can't send an e-mail to it, then the problem is *not* a networking problem, but an application problem (with the e-mail program).

exam Watch

The telnet application is used to test layer-7 (application layer) connectivity.

Using Telnet

If you've configured your Cisco devices correctly (with IP addressing and routing information), you should be able to successfully to telnet to them. Cisco routers and 2950 switches support both incoming and outgoing telnet, whereas the 1900 supports only incoming telnet. This section, and the following sections on telnet, are applicable only to Cisco routers and the 2950 Catalyst switches.

To open up a telnet session from a router or 2950 switch, you can use any of the following three methods:

```
Router# name_of_the_destination | destination_IP_address
-or-
Router# telnet name_of_the_destination | destination_IP_address
-or-
Router# connect name_of_the_destination | destination_IP_address
```

All three of these methods perform in the same manner: they all have the IOS attempt to telnet the specified destination.

Suspending Telnet Sessions

If you are on a switch or router and telnet to a remote destination, you might want to go back to the original switch or router. One way of doing this is to exit the remote device; however, you might just want to go back to your source Cisco device, do something real quick, and then return to the remote device. Logging out of and back into the remote device is a hassle, in this instance.

Cisco, however, has solved this problem by allowing you to *suspend* a telnet connection, return to your original router or switch, do what you need to do, and then jump right back into your remote device . . . all without having you to log out of and back into the remote device. To suspend a telnet session, use the CTRL-SHIFT-6 X or CTRL-^ control sequence, depending on your keyboard. You must hold down the CTRL, SHIFT, and 6 keys simultaneously, let go, and then hit the X key.

On your source router or switch, if you want to see the open telnet sessions that you have currently suspended, use the **show sessions** command:

```
Router# show sessions
Conn Host       Address    Byte   Idle    Conn Name
1 10.1.1.1   10.1.1.1      0      1    10.1.1.1
*  2 10.1.1.2   10.1.1.2      0      2    10.1.1.2
```

This example shows two open telnet connections. The one with the "*" preceding it is the default (last) session. To resume the last session, all you have to do is hit ENTER on an empty command line.

To resume a specific session, use this command:

```
Router# resume connection_#
```

The connection number to enter is the number in the "Conn" column of the **show sessions** command. If you are on the source router or switch and wish to terminate a suspended telnet session without having to resume the telnet session and then log out it, you can use this command:

```
Router# disconnect connection_#
```

Verifying and Clearing Connections

This section also deals with the router and 2950 switches. If you are logged into one of these devices, you can view the other users that are also logged in with this command:

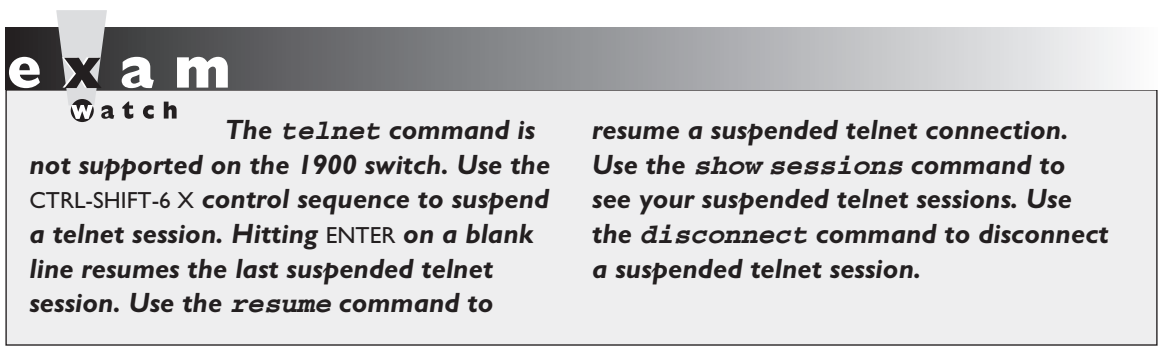
```
Router# show users
Line         User     Host(s)   Idle    Location
0    con 0                     idle
2    vty 0            idle        0    10.1.1.1
*   3    vty 1            idle         0    10.1.1.2
```

If you see a "*" in the first column, this indicates your current session. If you want to terminate someone's session, use the *Privilege EXEC* **clear line** command:

```
Router# clear line line_#
```

The line number that you enter can be found in the "Line" column of the output of the **show users** command.

6.17. The CD contains a multimedia demonstration of using telnet on a router.



The `telnet` command is not supported on the 1900 switch. Use the CTRL-SHIFT-6 X ***control sequence to suspend a telnet session. Hitting*** ENTER ***on a blank line resumes the last suspended telnet session. Use the `resume` command to resume a suspended telnet connection. Use the `show sessions` command to see your suspended telnet sessions. Use the `disconnect` command to disconnect a suspended telnet session.***

Debug Overview

One of the most powerful troubleshooting tools of the IOS is the **debug** command, which enables you to view events and problems, in real time, on your Cisco device. One problem of using **show** commands is that they display only what is currently stored somewhere in the router's RAM, and this display is *static*. You have to re-execute the command to get a refreshed update. And **show** commands, unfortunately, do not always display detailed troubleshooting information. For instance, maybe you want the router to tell you when a particular event occurs and display some of the packet contents of that event. The **show** commands cannot do this; however, **debug** commands can.

debug commands, however, do have a drawback: since the router has to examine and display many different things when this feature is enabled, the performance of the IOS will suffer. As an example, if you want to see every IP packet that travels through a router, the router has to examine each packet, determine if it is an IP packet, and then display the packet or partial packet contents on the screen. On a very busy router, this debug process can cause serious performance degradation. Therefore, you should be very careful about enabling a debug process on your router; you might want to wait till after hours or periods of inactivity before using these commands.

Also, you should never use the `debug all` command—this enables debugging for every process related to IOS features enabled on your router. In this situation, you'll just see pages and pages of output messages on all kinds of things and, on a busy router, probably crash it.

Typically, you will use **debug** commands for detailed troubleshooting. For instance, you may have tried using **show** commands to discover the cause of a particular problem, but without any success. You should then turn to using a particular **debug** command to uncover the cause of the problem. This command has many, many options and parameters—use the context-sensitive help to view them. Many of the remaining chapters in this book will cover specific debug commands and their uses. To enable debug, you must be at *Privilege EXEC* mode.

Once you've fixed your problem or no longer need to see the debug output, you should always disable the debug process. You can disable it by either prefacing the **debug** command with the **no** parameter or executing one of the following two commands:

```
Router# no debug all
Router# undebug all
```

These two commands disable all running **debug** commands on your router. You can first use the **show** `debug` command to see which events or processes you have enabled.

If you want to see timestamps displayed in your debug output, enter the following command:

```
Router(config)# service timestamps debug datetime msec
```

The **datetime** parameter displays the current date and time, while the **msec** parameter displays an additional timing parameter: milliseconds.

If you think your **debug** commands are causing performance problems, use the **show processes** command to check your CPU utilization for the device's various processes, including debug.

6.18. The CD contains a multimedia demonstration of using debug on a router.

Use the `undebug all` or `no debug all` command to disable all debug functions.

EXERCISE 6-2

Using the Router's Troubleshooting Tools

These last few sections dealt with the router's troubleshooting tools. This exercise will help you reinforce this material. You can find a picture of the network diagram in the Introduction of this book. You'll perform these steps using Boson's NetSim™ simulator. You can find a picture of the network diagram for the simulator in the Introduction of this book. After starting up the simulator, click on the *LabNavigator* button. Next, double-click on *Exercise 6-2* and click on the *Load Lab* button. This will load the lab configuration based on Chapter 5's exercises.

1. Access the 2600 router in the simulator on the CD. See what neighbors are directly connected to the router. What is the IP address of the 2500 router?

   At the top of the simulator in the menu bar, click on the *eRouters* icon and choose *2600*. Use **show cdp neighbors** command to view the 2600's neighbors—you may have to wait 60 seconds to see neighbors from this interface. You should see one of the 2950 switches and the 2500 router. Use the **show cdp neighbors detail** command to view the 2500's address: it is 192.168.2.2.

2. Access the 1900-1 switch in the simulator on the CD. See what neighbors are directly connected to the router. Which neighbors do you see? What are their IP addresses?

   At the top of the simulator in the menu bar, click on the *eSwitches* icon and choose *1900-1*. Use the **`show cdp neighbors`** command to view your neighbors. You should see the 2950-1 and 2950-2 switches. Add the **`detail`** parameter to the preceding command to see their IP addresses.

CERTIFICATION SUMMARY

The router contains the following components in ROM: POST, bootstrap program, ROM Monitor, and Mini-IOS. POST performs hardware tests; the bootstrap program finds and loads the IOS. ROM Monitor provides basic access to the router to perform testing and troubleshooting. The Mini-IOS is used in emergency situations when an IOS image cannot be located: it contains a stripped-down version of the IOS.

The configuration register affects how the router boots up. By default, POST is run, the bootstrap program is loaded, the IOS image is located, and the configuration file is executed. You can change this by using **`boot system`** commands or by changing the configuration register value. The **`show version`** command displays the current register value and what it will be upon a reload. The default register value is typically 0x2102. For the password recovery, use 0x2142.

Use the **`copy`** commands to manipulate files, including the configuration file and IOS images. Anytime you copy something into RAM, the IOS uses a merge process. For any other location, the IOS uses an overwrite process. On a 1900 switch, use the **`delete nvram`** command to delete the configuration file in NVRAM. On the 2950s and routers, its **`erase startup-config`**.

CDP is a Cisco-proprietary protocol that functions at the data link layer. Every 60 seconds, Cisco devices generate a multicast on each of their interfaces containing basic information about themselves, including the device type, the version of software they're running, and their IP address. To disable CDP globally, use the **`no cdp run`** command. To see a list of your neighbors, use the **`show cdp neighbors`** command.

The **`ping`** and **`traceroute`** commands support an extended version at *Privilege EXEC* mode. The 1900 doesn't support telnet, but almost all other IOS devices do. If you want to suspend an active telnet session, use the CTRL-SHIFT-6 X

control sequence. Hitting ENTER on a blank line resumes the last suspended telnet session. Use the **resume** command to resume a telnet connection. Use the **show sessions** command to see your open telnet session. Use the **disconnect** command to disconnect a suspended telnet session. To disable debug on your IOS device, use **undebug all** or **no debug all**. Debug functions only at *Privilege EXEC* mode.

TWO-MINUTE DRILL

Router Hardware Components and Bootup Process

- ROM stores the Mini-IOS, the bootstrap program, ROMMON, and POST. Flash stores the IOS images. NVRAM stores the configuration files. RAM stores the active configuration, including tables and buffers.
- When booting up, the router loads and runs POST from ROM. It then loads the bootstrap program from ROM, which, in turn, finds and loads the IOS. The IOS can be found in flash, TFTP, or ROM. The IOS then loads the configuration file, found in NVRAM.
- The configuration register and **`boot system`** commands can be used to override the default router bootup behavior. Use the **`show version`** command to see the register value. If the fourth hexadecimal character is 0x0, the router boots into *ROMMON* mode; if 0x1, the router boots the Mini-IOS; if 0x2–-0xF, the router uses the default boot sequence. The default configuration register value is 0x2102. For the password recovery, it's 0x2142.
- Here is the default bootup process: the bootstrap program examines the configuration register to determine how to boot up. If it is the default, the bootstrap program looks for **`boot system`** commands in the configuration file in NVRAM. If none are found, it looks for the IOS in flash. If no files are found in flash, then the bootstrap program generates a TFTP local broadcast to locate the IOS. If no TFTP server is found, the bootstrap program loads the Mini-IOS in ROM. If there is no Mini-IOS in ROM, the bootstrap program loads ROMMON.

Router Configuration Files and Flash

- These commands perform a merge process: **`copy startup-config running-config`**, **`copy tftp running-config`**, and **`configure terminal`**. These commands perform an overwrite process: **`copy running-config startup-config`** and **`copy running-config tftp`**.
- The 1900 automatically saves its active configuration to NVRAM, while the router and the 2950 switch require you to execute the **`copy running-config startup-config`** command.

- ❑ When upgrading your IOS, make sure you download the version of IOS from Cisco that contains the features that you purchased and verify that your router has enough flash and RAM for the new image. Use the **`copy tftp flash`** command to do perform an IOS upgrade.
- ❑ Use the **`reload`** command to reboot your router.

IOS Troubleshooting

- ❑ For layer-2 troubleshooting, use the **`show interfaces`** command and CDP. For layer-3 troubleshooting, use **`ping`** and **`traceroute`**. For layer-7 troubleshooting, use telnet. For detailed troubleshooting, use **`debug`**.
- ❑ CDP is used to learn basic information about directly connected Cisco devices. It uses a SNAP frame format and generates a multicast every 60 seconds. It is enabled, by default, on a Cisco device.
- ❑ To execute an extended **`ping`** or **`traceroute`**, you must be at *Privilege EXEC* mode. Ping tests only if the destination is reachable, while **`traceroute`** lists each layer-3 device along the way to the destination.
- ❑ To suspend a telnet session, use the CTRL-SHIFT-6 X or CTRL-^ control sequence.
- ❑ The **`debug`** commands require *Privilege EXEC* access. To disable all **`debug`** commands, use **`no debug all`** or **`undebug all`**.

SELF TEST

The following Self Test questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Choose all correct answers for each question.

Router Hardware Components

1. Which of the following is stored in ROM? (Choose all correct answers.)
   - A. POST
   - B. ROMMON
   - C. Configuration file
   - D. System recovery file
2. Which types of memory do not maintain their contents during a power-off state?
   - A. NVRAM
   - B. ROM
   - C. RAM
   - D. Flash

Router Bootup Process

3. The _____ program goes out and finds the IOS and loads it.
   - A. ROMMON
   - B. Bootstrap
   - C. Mini-IOS
   - D. Loader
4. Enter the router *Configuration* command to have it boot up from the Mini-IOS in ROM:
   _____
5. Which router command would you use to view the configuration register value?
   - A. `show register`
   - B. `show interfaces`
   - C. `show configuration`
   - D. `show version`

6. Enter the router *Configuration* mode command to change the configuration register to 0x2142: _____.

Router Configuration Files

7. Which router command performs an overwrite process? (Choose all correct answers.)
   A. `copy running-config startup-config`
   B. `copy startup-config running-config`
   C. `copy tftp running-config`
   D. `copy running-config tftp`
8. Enter the router command to delete your configuration file in NVRAM: _____.

Router IOS Image Files

9. IOS images can be loaded from all the following except:
   A. ROM
   B. Flash
   C. NVRAM
   D. TFTP
10. When backing up your IOS image from flash, which of the following will the **`copy flash tftp`** command prompt you for? (Choose all correct answers.)
    A. TFTP server IP address
    B. Verification to copy
    C. Source filename
    D. Destination filename

IOS Troubleshooting

11. Which router command would you use to test only layer-3 connectivity?
    A. `telnet`
    B. `show cdp traffic`
    C. `show interfaces`
    D. `traceroute`

12. Extended **`traceroute`** works from which mode?

A. *User EXEC*
B. *Privilege EXEC*
C. *Configuration*
D. *User* and *Privilege EXEC*

13. How would you suspend a telnet session?

A. CTRL-SHIFT-X 6
B. CTRL-SHIFT-6 X
C. CTRL-6 X
D. CTRL-C

14. Which router command would take you back to a suspended telnet session?

A. `reconnect`
B. `connect`
C. `resume`
D. `toggle`

15. Enter the router command to disable all debug processing: _____.

SELF TEST ANSWERS

Router Hardware Components

1. ☑ **A.** and **B.** POST, ROMMON, the Mini-IOS, and the bootstrap program are in ROM.
   ☒ **C** is stored in NVRAM, and **D** is nonexistent.
2. ☑ **C.** RAM contents are erased when you turn your device off.
   ☒ **A**, **B**, and **D** are incorrect; ROM, NVRAM, and flash maintain their contents when the device is turned off.

Router Bootup Process

3. ☑ **B.** The bootstrap program goes out and finds the IOS and loads it.
   ☒ **A** is incorrect because it applies to *Monitor* mode. **C** is incorrect because that is the IOS, though it may be a stripped-down one with only IP included. **D** is a nonexistent process.
4. ☑ Use the **`boot system rom`** or **`config-register 0x2400`** command to boot the Mini-IOS in ROM.
5. ☑ **D.** Use the **`show version`** command to view your configuration register value.
   ☒ **B** shows only interface statistics, and **A** is a nonexistent command. **C** is the old command version for **`show startup-config`**.
6. ☑ Enter the **`config-register 0x2142`** command in *Configuration* mode. This will cause the router to boot up and not load the configuration file in NVRAM.

Router Configuration Files

7. ☑ **A** and **D.** Copying to any other place besides RAM causes an overwrite.
   ☒ **B** and **C** are wrong because copying to RAM is a merge process, not an overwrite process.
8. ☑ **A.** Use the **`erase startup-config`** command to erase your configuration file in NVRAM.

Router IOS Image Files

9. ☑ **C.** NVRAM stores configuration files, not IOS images.
   ☒ **A, B,** and **D** are incorrect because IOS images can be loaded from ROM (the Mini-IOS, flash (the default), or a TFTP server.

10. ☑ **A., C.,** and **D.** When you use the **`copy flash tftp`** command, you are prompted for the TFTP server's IP address, the source filename of the IOS in flash, and the name you want to call the IOS image on the TFTP server.
☒ **B** is incorrect because you didn't are not prompted for a verification before the command is executed.

IOS Troubleshooting

11. ☑ **D.** The **`traceroute`** command tests layer-3.
☒ **A** is incorrect because it tests layer 7. **B** and **C** are incorrect because they test layer 2.

12. ☑ **B.** You need to be at *Privilege EXEC* mode to use extended **`ping`** and **`traceroute`**.
☒ **C** is incorrect because you can only make configuration changes here. **A** and **D** are incorrect because they include *User EXEC* mode.

13. ☑ **B.** Use CTRL-SHIFT-6 X to suspend a telnet session.
☒ **D** is incorrect because this break sequence is used to break out of the *System Configuration Dialog*. **A** and **C** are incorrect because these are nonexistent break sequences.

14. ☑ **C.** Use the **`resume`** command to reconnect to a suspended telnet session. You can also hit the ENTER key on a blank line to return to the last suspended telnet session.
☒ **B** is incorrect because this performs a telnet. **A** and **D** are incorrect because these are nonexistent commands.

15. ☑ Use **`no debug all`** or **`undebug all`** from *Privilege EXEC* mode to disable all debug processing**.**