Exam : 350-018(LAB)

Title : Network Security

Ver : 12.19.06

- **Network Security**

Lab Abstract

Currently your routers have no security features configured on them beyond enable secret passwords and login passwords on the vty lines for Telnet access. All of the vty lines share the same password. You decide to implement some security features. Here is what you want to do:

1. Configure each of the routers with passwords for Console access.
2. "Reserve" one vty line on each router for your own access by setting a different password on it.
3. Change the enable secret password on all the routers.
4. Configure access lists on each router to allow Telnet connections only from your workstation (IP address 172.18.56.14)
5. Configure access lists on each router to deny all ping requests sent to the routers from workstation (IP address 172.18.56.16)
6. Log any traffic that is denied by the access lists that you implement.
7. Make sure that no other network traffic is impacted by the implementation of these access lists.

Lab Scenario

Network Security Lab Scenario

Introduction

You are the network administrator for The Meely Meal company. Owned by Milton Meely, the company is a leading distributor of wheat germ and other grains and cereals. The company has three locations:

1. Corporate Headquarters in Albuquerque, New Mexico.

2. A packaging and distribution plant in Battle Creek, Michigan.

3. A small purchasing office in Lincoln, Nebraska.

A diagram of the network is included below.

Milton has hired his son, Matt, as an intern for the summer. Matt tells you that he is thinking of getting his CCNA. He says that he plans to prepare by reading "the" book. You tell him that it might be a good idea to get some hands on experience before taking the test. Milton thinks is a great idea. Suddenly Matt is your new "assistant" and wants to have access to the company routers so he can play with them.

Needless to say, you are concerned, and you want to limit the access that he has. You are willing to teach him IOS commands as long as you are standing with him while he connects to the local router through the console port, but you do not want him accessing the routers remotely while you are not around.

Currently the routers have no security features configured on them beyond enable secret passwords and login passwords on the vty lines for Telnet access. All of the vty lines share the same password. Here is what you want to do:

Objectives

1. Configure each of the routers with passwords for Console access.

2. "Reserve" one vty line on each router for your own access by setting a different password on it.

3. Change the enable secret password on all the routers.

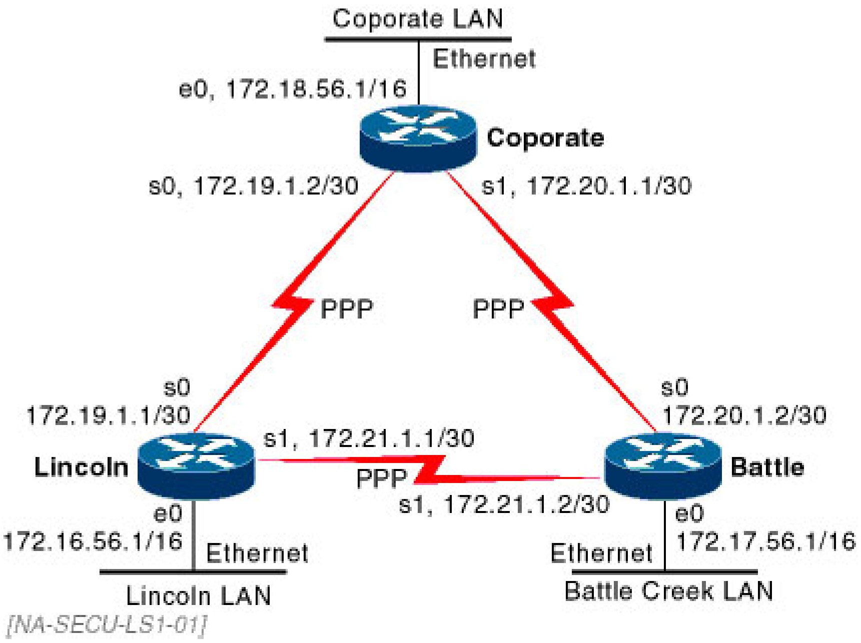
4. Configure access lists on each router to allow Telnet connections only from your workstation (IP address 172.18.56.14).

5. Configure access lists on each router to deny all ping requests sent to the routers from Matt's workstation (IP address 172.16.56.16).

6. Log any traffic that is denied by the access lists that you implement.

7. Make sure that no other network traffic is impacted by the implementation of these access lists.

Network Diagram



Solution

1. Login to each router and enter Privileged Exec mode. Enter Global configuration mode with the **configure terminal** command. Use the **line con 0** command to configure the console line. Use the **login** and **password** commands to configure the console for login with a password. Here is an example using the Battle Creek router:

```
Battle>enable
Password:*******
Battle#conf term
Battle(config)#line con 0
Battle(config-line)#login
Battle(config-line)#password oatmeal
Battle(config-line)#^Z
```

2. While logged into the router, enter Privileged Exec mode. Then enter Global Configuration mode. Use the **line vty** command to configure the virtual terminal lines. First configure lines 0 through 3 using the **line vty 0 3** command. Assign a password to these four lines. Then configure the last line with a different password using the **line vty 4** command. Here is an example on the Battle Creek router:

```
Battle>enable
Password:*******
Battle#conf term
Battle(config)#line vty 0 3
Battle(config-line)#login
Battle(config-line)#password oatbran
Battle(config-line)#^Z

Battle#conf term
Battle(config)#line vty 4
Battle(config-line)#login
Battle(config-line)#password shellfish
Battle(config-line)#^Z
```

3. Connect to the router, and enter Global Configuration mode. Use the **enable secret** command to change the enable secret password. Here is an example:

```
Battle>enable
Password:*******
Battle#conf term
Battle(config)#enable secret wheatgerm
Battle(config)#^Z
```

4,5,6, and 7. Configure an Extended IP access list on each router that first permits the desired traffic, then denies the undesired traffic, then permits all other traffic. Make sure you end each access list entry with the log keyword. Assign the access list as an incoming filter on each of the routers' serial interfaces with the **ip access-group in** command. Here is an example of the procedure:

```
Battle>enable
Password:*******
```

```
Battle#conf term
Battle(config)#no access-list 101
Battle(config)#access-list 101 permit tcp host 172.18.56.14 ...
                                          any eq telnet log
Battle(config)#access-list 101 deny tcp any any eq telnet log
Battle(config)#access-list 101 deny icmp host 172.18.56.16 ...
                                   any eq echo-request log
Battle(config)#access-list 101 permit ip any any
Battle(config)#int s0
Battle(config-int)#ip access-group 101 in
Battle(config-int)#int s1
Battle(config-int)# ip access-group 101 in
Battle(config-int)#^Z
```

The access list above does the following:

- Line 1 allows Telnet connections from the host IP address of 172.18.56.14.

- Line 2 drops all other Telnet traffic (Lines 1 and 2 meet lab objective #4).

- Line 3 drops ping requests from the host IP address of 172.18.56.16 (lab objective #5).

- Line 4 allows all other traffic to pass (meeting objective #7).

- All lines end with the log keyword (meeting objective #6).

Router Configurations

Corporate Router

```
!
!
hostname Corporate
!
enable password wheatgerm
!
no ip name-server
!
ip routing
!
access-list 101 permit tcp host 172.18.56.14 any eq telnet log
access-list 101 deny tcp any any eq telnet log
access-list 101 deny icmp host 172.18.56.16 any eq echo-request log
access-list 101 permit ip any any
!
interface Ethernet 0
 no shutdown
 description connected to Corporate LAN
 ip address 172.18.56.1 255.255.0.0
 keepalive 10
 ip access-group 101 in
!
```

```
interface Serial 0
 no shutdown
 description connected to Lincoln
 ip address 172.19.1.2 255.255.255.252
 encapsulation ppp
!
interface Serial 1
 no shutdown
 description connected to Battle
 ip address 172.20.1.1 255.255.255.252
 encapsulation ppp
!
router rip
network 172.18.0.0
 network 172.19.0.0
 network 172.20.0.0
 no auto-summary
!
!
!
line console 0
 exec-timeout 0 0
 password oatmeal
 login
!
line vty 0 3
 password oatbran
 login
!
line vty 4
 password shellfish
 login
!
end
```

Battle Creek Router

```
!
service timestamps debug uptime
service timestamps log uptime
!
hostname Battle
!
enable password wheatgerm
!
no ip name-server
!
ip subnet-zero
no ip domain-lookup
ip routing
!
access-list 101 permit tcp host 172.18.56.14 any eq telnet log
access-list 101 deny tcp any any eq telnet log
access-list 101 deny icmp host 172.18.56.16 any eq echo-request log
access-list 101 permit ip any any
!
```

```
interface Ethernet 0
 no shutdown
 description connected to Battle Creek LAN
 ip address 172.17.56.1 255.255.0.0
 keepalive 10
!
interface Serial 0
 no shutdown
 description connected to Corporate
 ip address 172.20.1.2 255.255.255.252
 encapsulation ppp
 ip access-group 101 in
!
interface Serial 1
 no shutdown
 description connected to Lincoln
 ip address 172.21.1.2 255.255.255.252
 encapsulation ppp
 ip access-group 101 in
!
router rip
network 172.17.0.0
 network 172.20.0.0
 network 172.21.0.0
 no auto-summary
!
!
!
line console 0
 exec-timeout 0 0
 password oatmeal
 login
!
line vty 0 3
 password oatbran
 login
!
line vty 4
 password shellfish
 login
!
end
```

Lincoln Router

```
!
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname Lincoln
!
enable password wheatgerm
!
no ip name-server
!
```

```
ip subnet-zero
no ip domain-lookup
ip routing
!
access-list 101 permit tcp host 172.18.56.14 any eq telnet log
access-list 101 deny tcp any any eq telnet log
access-list 101 deny icmp host 172.18.56.16 any eq echo-request log
access-list 101 permit ip any any
!
interface Ethernet 0
 no shutdown
 description connected to Lincoln LAN
 ip address 172.16.56.1 255.255.0.0
 keepalive 10
!
interface Serial 0
 no shutdown
 description connected to Corporate
 ip address 172.19.1.1 255.255.255.252
 encapsulation ppp
 ip access-group 101 in
!
interface Serial 1
 no shutdown
 description connected to Battle
 ip address 172.21.1.1 255.255.255.252
 encapsulation ppp
 ip access-group 101 in
!
router rip
 version 2
 network 172.16.0.0
 network 172.19.0.0
 network 172.21.0.0
 no auto-summary
!
!
!
line console 0
 exec-timeout 0 0
 password oatmeal
 login
!
line vty 0 3
 password oatbran
 login
!
line vty 4
 password shellfish
 login
!
end
```

Tutorial

Access Denied: Network Security with Cisco Routers for the CCNA candidate

Introduction

Preparing for the CCNA exam is sometimes like eating at one of those "all you can eat" restaurants. There is an abundance of study material to choose from, and it can be difficult to decide what you really need. If you spend your whole time in the restaurant eating fried clams, you won't get the nutrition that you needed from that meal and you'll probably end up with a stomachache. Likewise, if you spend all of your study time reading books about TCP/IP, you will not get enough information to pass the exam and you'll probably wind up with a headache. A good way to make sure that you get the nutrition that you need from eating is to create a meal plan, with all of the foods selected to give you a nutritionally balanced meal. Then eat only the foods listed in the plan, and only the amounts of each food that is specified in the plan.

The same approach can be taken when getting ready to take the CCNA exam. Consider the exam objectives posted online by Cisco to be your CCNA meal plan. Study the material covered by the objectives listed. Some material is stressed more heavily than other material on the exam, and that is reflected in the objectives. Some concepts are weighted heavily on the exam, and consequently appear in the objective list frequently. Other concepts are not covered much by the exam, and they appear only once or twice in the objective list. An example of this is the Network Security section of the CCNA objectives list. Notice that there are only two objectives under this heading. This would suggest that network security is not stressed too heavily on the exam, and in fact, it is not.

But, do not think that you need not study this material! You do. The point is that you need to focus your study time on the material covered by the objectives listed and not waste time with other tangential material. The CCNA objective list is perhaps not as well organized as it might be. A close examination of the list shows that some material that could be listed under the Network Security section is listed elsewhere instead. In this paper, I will try to pull this material together and give you the information that you will need to know to answer the questions you will see on the exam that relate to the following objectives:

- c1) Log into a router in both user and privileged mode

- c6) Control router passwords, identification, and banner

- f1) Configure standard and extended access lists to filter IP traffic

- f2) Monitor and verify selected access list operations on the router

Keep in mind that network security is an advanced discipline and really has a career path of its own. The material presented here will help you to pass the CCNA exam, but there is certainly a smorgasbord of additional network security material available. Now, let's get on to the meat and potatoes!

A very brief mention of security principles may help get us started. Much more discussion of this material is available in some of the references at the end of this paper.

Accessing real or virtual consoles, in formal security terms, is a problem of *authentication*: determining that the purported user is actually who he or she claims to be. Most authentication systems are what security experts call *two-factor*, the two factors being who you are and something only you know or have. "Who you are" is your user ID, while the password is something you know. Other second factors include one-time passwords from smart token cards or password lists, or biometric identifiers such as fingerprint or retinal scanners.

The general routing access lists described in this paper are means of access control: permitting or denying traffic based on certain criteria. Access control lists, in general terms, consist of a pattern to match and an action, such as permitting flow, that takes place when the pattern is matched.

A broader industry term is Access Control Lists, or ACLs. ACLs are available on both hosts and routers. A complete security solution will use them in both places, as well as host-level authentication and other security functions.

The Router Console

Most Cisco routers run the Cisco IOS software to perform all of their functions. The IOS interface with which you interact is called Exec or the Command Line Interpreter (CLI). This is the command interpreter that accepts your configuration commands and acts upon them. You can access the Exec command line in a number of ways: through the console port, through a modem connected to the auxiliary port, or through a virtual terminal session on one of the router's appropriately configured network interfaces.

IOS Exec Modes

Cisco IOS operates at different levels called Exec Modes. Each mode allows you to perform certain tasks. Once you gain access to the Exec command line, you can perform some actions on the router, such as view the version of the IOS software running on the router or look at the router's running configuration. To perform other actions, such as change the configuration of the router, you must be operating in a different Exec mode. The two Exec modes that you will be most concerned with are User Mode and Privileged Mode.

User Mode is the Exec mode that you are in when first accessing the Exec command line. It allows you to use a limited subset of the IOS commands. To view the commands available to you in User Mode, simply type the following at the command prompt:

```
Router>?
```

When you press enter, you will see a list of the IOS commands that are available to you in User Mode. One command will be of particular interest to you. That command is **enable**. The **enable** command is used to enter the next level of IOS privilege, called Privileged Mode.

In Privileged Mode, you have considerably more access to the router. You can access more system information and operating statistics and you can change the global configuration of the router. From Privileged Mode, you can configure the individual interfaces on the router as well as each of the protocols that the IOS software is configured to support. To see a list of the IOS commands available in Privileged Mode, first use the **enable** command to enter Privileged Mode and then enter the help command (**?**) as you did earlier in User Mode:

```
Router>enable
Password:*******
Router#?
```

This time when you type the question mark and press enter, you will see many more commands listed than you did when you performed this exercise in User Mode. From Privileged Mode you can enter other Exec modes, like Global Configuration Mode or Interface Configuration Mode.

From a security standpoint, it is desirable to control access to the Exec command line itself, as well as to the Privileged Exec Mode. This is done in a number of ways. Notice that when you used the **enable** command in the example above, you were prompted for a password. The password protects access to the Privileged Mode on the router.

The password for Privileged Mode is set using either the **enable password** command or the **enable secret** command. The **enable secret** command is used to create an encrypted password for access to Privileged Mode. The **enable secret** password is used by IOS versions 10.3 and above, and is preferred over the **enable password** password when both are configured.

The enable password command is used to configure a password for access to Privileged Mode. The password is not encrypted unless you do so manually. This command is used in IOS versions earlier than 10.3 as the only method to configure a Privileged Mode password. With version 10.3 and higher of the IOS, this command will configure a password that will be used when no **enable secret** password has been configured. You can configure the **enable password** password and the **enable secret** password to be the same, but IOS will give you a warning when you do so. You can ignore this warning if you really want the two passwords to be the same.

Both the **enable password** and the **enable secret** commands are Privileged Mode commands. The router must be in Global Configuration Mode (accessed through Privileged Mode) for these commands to execute successfully. Of course, if no password has been previously configured, it should be no problem to get to the right Exec mode. The router's Initial Configuration Dialogue, if used, will prompt you to set up a Privileged Mode password. Once a Privileged Mode password has been configured, you will need to know what it is in order to change it to something else. Lost or forgotten passwords can be a bit of a pain for router administrators. Password recovery, though beyond the scope of this paper, is well documented elsewhere for various IOS versions and different Cisco hardware. It is well worth the effort to learn password recovery techniques for the Cisco devices that you must support.

Interpreting Prompts

Notice that the Privileged Mode Exec prompt is a # sign. This is different from the User Mode > sign. Pay particular attention to the prompt to determine in which IOS mode you are operating. Some of the questions on the exam may be designed to catch your attention to detail in this regard. Watch out for answers that appear correct because they have the correct command syntax but are actually incorrect because the wrong Exec mode prompt is displayed. For example, the following is an invalid command line:

```
Router>debug ip rip
```

You cannot execute the **debug ip rip** command in User Mode. You must be in Privileged Mode for this command to execute successfully. The correct prompt would appear as follows:

```
Router#debug ip rip
```

It is easy to overlook this type of thing under the pressure of a live exam. Remember to take your time, stay calm, and read each answer carefully and you will be sure to spot these kinds of detractors.

Securing Console Access

Once you have controlled access to the Privileged Mode with a password, you will want to control access to the Exec command line itself. You will need to consider keeping the router in a secure location to control access to the console port itself, and then use the **line con 0** command to configure the console port for a login password. Follow these steps to configure the console port to prompt for a password before allowing access to the Exec command line:

```
Router>enable
Router#config term
Router(config)#line con 0
Router(config-line)#login
Router(config-line)#password cisco
Router(config-line)#^Z
```

Securing Modem Access

Additionally, you will want to configure a login password for access to the Exec command line through remote means. If you have a modem connected to your aux port, you will need to configure an auxiliary port password to control access to the router through that port. To do this, you must first configure the aux port using the **line aux 0** command. This command is a Global Configuration Mode command that allows you to configure the first auxiliary port (port 0). Follow these steps to get to the correct mode and configure the aux port with a password:

```
Router>enable
Password:*******
Router#config term  ; puts you in Global Configuration Mode
Router(config)#line aux 0
Router(config-line)#login  ; Now in Line Configuration Mode
Router(config-line)#password cisco
Router(config-line)#^Z  ; saves changes, exits Config Mode
```

Note: Do not actually set the password to "cisco" -- use a password that is more difficult to guess.

Securing Telnet Access

In addition to setting a password for the aux port, you will want to setup access to the Exec command line through Telnet sessions to virtual terminal lines on the router. These virtual terminal (vty) lines allow you to connect to the router through Telnet sessions to its network interfaces. The network interfaces must have the IP protocol configured to support Telnet sessions. The router must also have its vtys configured, and you must setup a vty password before the router will accept any incoming Telnet sessions. To do this, use the **line vty 0 4** command as follows:

```
Router>enable
Password:*******
Router#config term
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#password cisco
Router(config-line)#^Z
```

Cisco routers can accept up to 5 Telnet sessions (numbered 0 through 4) concurrently. The vtys for all of these sessions are configured with **line vty 0 4** command. The 0 and the 4 in the command indicate the first and last session configured by the line vty command. You can configure each line individually by only specifying one vty number in the command. For example, to configure only the fourth vty line, use the following command:

```
Router(config)#line vty 4
```

It is common practice to configure at least one vty with a different password from the others and to limit who has access to this vty password so that there will always be an available vty when needed.

Router Identification

In the configuration examples used so far, you will notice that the prompt always begins with the word "Router." This is the actual name of the router that we are configuring. If we want to change that name, we use the **hostname** command. It is a good idea to use a host name that is meaningful to anyone who needs to administer the router, but that does not give away too much information to someone who accesses the router without appropriate authority to do so. You might want the router's host name to indicate the location of the router, or the router's role in the internetwork, but try to do so using some sort of naming convention rather than stating it explicitly in the host name. For example, try the name **fl3ar1** instead of the name **3rdFloorAccessRouter1**. This may be considered a small point, but it is part of the overall security of the internetwork. Each small component works with each other to form the composite security architecture -- in other words, every little bit counts.

To configure the host name on a router, follow these steps:

```
Router>enable
Router(config)hostname fl3ar1
Router(config)^Z
fl3ar1>
```

Banner Messages

Another small component in the overall network security architecture is the router's banner. The banner is a message that the router displays whenever you attempt to access the Exec command line. It might be tempting to place a banner message that says something like "Welcome to the company's Cisco internetwork. Call the help desk at 555-1212 for support." This is not a very good banner message, from a security perspective. There have been cases where administrators have attempted to prosecute people who have accessed their internetworks illegally, only to find that the case shot down by the perpetrator's claim that the company's banner message gave them the impression that they were welcome there. A better banner message might be one that indicates that unauthorized access will be prosecuted. Try something along the lines of the following:

"You have accessed a private internetwork. Unauthorized access to this internetwork is prohibited and will be prosecuted in accordance with Title 18, U.S.C. -- if you are not explicitly authorized to access this internetwork, log off now!"

To configure the router's banner, use the **banner motd** command. What the heck is "motd" you ask? It is short for **m**essage **o**f **t**he **d**ay. In this particular case, unless you change it yourself daily, it is more like the message of **every** day. To use this command successfully, you must specify a delimiter (of your choice) that indicates the end of your message. It is common to use the # sign as a delimiter. Here is an example:

```
Router>enable
Router#config term
Router(config)#banner motd #
Enter TEXT message. End with the character '#'.
You have accessed a private internetwork.
Unauthorized access to this internetwork
is prohibited and will be prosecuted in
accordance with Title 18, U.S.C. -- if you
are not explicitly authorized to access
this internetwork, log off now!
#
Router(config)#^Z
```

Configuring a banner message like the one above will provide an effective indicator regarding who is allowed to access your internetwork to those who connect to your router either deliberately or accidentally.

At this point, we have covered all of the material in the first two objectives we listed in the introduction. This material is fairly simple and straightforward. Now let's move on to material that is a bit more complex.

Access Lists

Password security is one level of an overall approach to securing your internetwork. Passwords are simple, but are also a weak level of security. Passwords are often easy to guess, and even the most complex of passwords can be derived given enough time. To take security to the next level, you will want to limit access to the router on a per packet basis. To accomplish this on Cisco routers, you use access lists.

Access lists are not used for security alone. They also perform other useful functions on Cisco routers whenever certain traffic must be identified on a per-packet basis. Access lists are useful when you need to filter traffic off low-bandwidth links or to trigger certain events such as the initiation of a dial-up connection to another router. This paper will focus primarily on the security uses of access lists, since access lists are specified in the Network Security section of the CCNA exam objectives, although the basic principles for creating access lists are the same regardless of what you are trying to accomplish with them.

An access list is a list of criteria used for identifying certain traffic, along with instructions for what action to take when this particular traffic is found. When access lists are executed, traffic is compared to certain patterns specified in the access list. When a match is found, then an action is taken as specified by the access list. The action to take is either to permit the traffic (allow the packet to pass through the router) or to deny it (drop the packet). Access lists are most often made up of a number of patterns for comparison, and packet information is compared to each one in turn until either a match is found, or the end of the list is reached. Before looking at the actual IOS syntax for creating access lists, let's examine how they work in general. Here is an example of an access list:

Criteria to compare	**Action to take**
Traffic destined for network X?	Deny
Traffic destined for host Y?	Permit
Traffic originating from host Z?	Permit
Traffic using port N?	Deny

In the preceding table is a list of criteria against which each packet will be compared. If the information in the packet matches the criteria in the comparison, then the associated action will be taken. Using the table, if a packet contains a destination address on network X, then it will be denied (the packet will be dropped). No further comparisons will be made for this packet; it will simply be dropped because it matched the first line in the list. If the packet is NOT destined for network X, then it will be compared to the criteria specified in the next line. In this case, if the packet is not destined for network X, the router will determine if the destination address is that of host Y. If it is, then the packet will be allowed to flow through the router. Again, no more comparisons will be made. **Once a packet matches a line in the access list, the corresponding action is taken, and no further comparisons are made.**

It is very important to remember the point above because it makes the order in which you specify criteria in your access list critical. Consider the following four hosts:

Hostname	Address
Arthur	network 1, host 1
Dipsy	network 1, host 2
Kermit	network 2, host 1
Kipper	network 3, host 1

Assume that you have a router that connects network 1, network 2, and network 3, and that you are configuring an access list on that router. The access list will filter traffic based on the following criteria:

Criteria to compare	Action to take
Traffic destined for network 1?	Deny
Traffic destined for host Dipsy?	Permit
Traffic destined for host Kipper?	Permit
Traffic destined for network 3?	Deny

Read the access list criteria in the table above, and pay attention to the order in which they are specified. Can traffic from network 2 or network 3 reach the host named Dipsy? The second line explicitly permits traffic destined for host Dipsy, so you might think that traffic could reach Dipsy. The fact is, though, that no traffic from networks 2 and 3 can reach the host Dipsy. Dipsy resides on network 1. Traffic destined for Dipsy will, of course, also be destined for network 1 in order to reach Dipsy. Any traffic destined for network 1, including traffic destined for Dipsy, will match line one of the access list. The action specified by line one is Deny, so the traffic will be dropped. **No further comparisons will be made for these packets!** The traffic destined for Dipsy will never be compared to the criteria specified in line two, because it will already have been dropped by matching line one. When creating access lists, the order in which you enter your criteria is critical to the effects of the access list.

What happens when a packet does not match any of the criteria specified in an access list? That is a very good question. If a packet should make it past the last line of an access list and not match any of the comparison criteria, the router needs to know whether to permit or deny it. The safest option, from a security perspective, is to deny it, and that is what Cisco routers will do. There is a term for this default action at the end of an access list -- "implicit deny any." It is never actually stated in the access list or printed in any configuration (hence the "implicit" in the name), but it is a part of every access list on a Cisco router. The "implicit deny any" ensures that any packet that does not match some explicitly stated access list criteria will be dropped once it passes the end of the list.

It is important to remember the "implicit deny any." If it is your intention to allow any traffic that does not meet any of your stated criteria, you must add to the access list a line that explicitly permits this traffic. This line will be the last line of the access list, and will permit the traffic before it is dropped by the "implicit deny any."

Types of Access Lists

Access lists are protocol specific. There are access lists for many different protocols, including TCP/IP, IPX/SPX, AppleTalk, DECnet, and Banyan VINES. The CCNA exam will focus mainly on the access lists that are used to filter TCP/IP traffic, although there may also be some questions on IPX/SPX access lists as well.

Standard and Extended IP access lists

IP access lists fall into two categories: standard IP access lists and extended IP access lists. Standard IP access lists are able to examine the source address of an IP packet and take action based on the information found there. Extended IP access lists offer much more flexibility. They can take action based upon a number of different fields in the IP packet, including the source address, the destination address, and the port number.

Access lists are identified by number. The access list number also indicates the type. Standard IP access lists, for example, are identified by a number within the range 1 to 99. Extended IP access lists are numbered from 100 to 199. This convention establishes a limit of 99 Standard IP access lists and 100 Extended IP access lists on the router. This limit can be overcome using named access lists, which I will discuss later in this paper.

The **access-list** IOS command is used to create an access list. The syntax for a Standard IP access list is as follows:

```
access-list number {deny|permit} source [source-wildcard]
```

Each line you configure for a Standard IP access list takes this form. The access-list command indicates that you are creating an access list. The "number" parameter uniquely identifies the access-list on the router and indicates its type. The {deny|permit} parameter indicates the action to take when a match is found, and the "source" parameter indicates the criteria for comparison (in this case, a source IP address).

The [source-wildcard] indicates a mask to apply to the source IP address in order to specify which bits in the address you care about matching. This mask, called the wildcard mask, allows you some granularity in specifying a match based on the source IP address. You can specify that you want all of the bits to match, indicating a specific host address, or you can specify that only certain bits need match, broadening the scope of your filter to a range of host addresses or subnet addresses.

Wildcard masks are often confused with IP subnet masks. They are similar in function, but opposite in approach. When an IP subnet mask is "applied" to an IP address, all the bits in the address that correspond to bits in the mask containing ones are considered part of the network portion of the address, while the bits in the IP address that correspond to the bits in that mask that contain zeros are considered part of the host address. With a wildcard mask, any bit in the IP address that corresponds to a zero bit in the mask must match the bit in the access list criteria exactly. Any bit in the IP address that corresponds to a one bit in the mask will match the bit in the access list criteria, regardless of its value.

Hint

For each octet, the sum of the equivalent subnet and wildcard masks should be 255. For example, for the subnet mask 255.240.0.0:

```
 255.240. 0 . 0  subnet mask
+ 0 . 15.255.255 wildcard mask
 ---------------
 255.255.255.255
```

Here is an example to make this clear:

```
access-list 1 permit 10.10.10.10 0.0.0.0
```

In this example, the source IP address is 10.10.10.10, and the wildcard mask is 0.0.0.0. This means that every bit in the mask is a zero. In this case, every bit in the source IP address of the packet being examined must match the bits in the address 10.10.10.10 in order for this line of the access list to be matched. This line effectively filters traffic from a specific host: 10.10.10.10, and allows it to pass through the router (thanks to the "permit" parameter). If I wanted to allow traffic originating from any host whose address contained 10 in the first three octets, I would use the following line:

```
access-list 1 permit 10.10.10.0 0.0.0.255
```

Notice that we have changed the wildcard mask so that now the last octet contains all ones. In this case, if a packet is examined, and each of the first three octets of the source IP address contained the value 10, the packet would be permitted, regardless of the value of the last octet. This would obviously allow much more traffic to pass through the router than the first example. You may also have noticed that we have changed the "source" parameter to 10.10.10.0 -- the zero at the end is really only a placeholder at this point. Since the entire last octet of the wildcard mask is set to ones, any value in the last octet will be declared a match. It makes no difference what we set this value to in the "source" parameter, but it is conventional to set it to 0 in this case. We really don't care what the value is, and in fact the one bits of a wildcard mask are often called the "don't care" bits.

Suppose that I want to deny traffic from every host on the 10.0.0.0 network. The following line would apply:

```
access-list 1 deny 10.0.0.0 0.255.255.255
```

In this case we don't care what the host portion of the address is, so we can set those bit positions in the wildcard mask to ones. We want to exactly match the value 10 in the first octet, so we set the first octet in the wildcard mask to all zeros.

Consider the following line:

```
access-list 1 permit 0.0.0.0 255.255.255.255
```

Which traffic will this line permit? Examine the source/source wildcard pair. The IP address of 0.0.0.0 is really meaningless in this case, because the wildcard mask contains a one in every position. What we are effectively specifying is that absolutely **any** address will match this line. Cisco has created a shortcut that you can use when you want to specify this address/wildcard pair. The shortcut is to use the **any** keyword. Using the shortcut we can rewrite the line like this:

```
access-list 1 permit any
```

This is a useful statement to put in an access list as the last line if you want to prevent traffic from being dropped by the "implicit deny any."

Let's put a few lines together to make a more functional access list. Look at this example:

```
access-list 1 deny 10.10.10.10 0.0.0.0
access-list 1 permit 10.10.10.0 0.0.0.255
access-list 1 deny 10.10.0.0 0.0.255.255
access-list 1 permit any
```

Read the Fine Print

One thing that you will notice as you look at access list lines is that it can become difficult to distinguish the IP address criteria from the wildcard mask. The numbers and the dots tend to blend together. The only thing that keeps the source/source wildcard pair 10.10.10.0 0.0.0.255 from being one long string is the missing period between the source IP address and the wildcard mask.

Be careful when taking the exam that you read the lines of the access lists carefully. Make sure that you are certain what the mask really is before you select an answer to an access list question. It can be easy to think that a zero at the end of the "source" parameter is actually a zero in the first octet of the wildcard mask.

This is one of those attention-to-detail issues again, but it can be a source of frustration for test takers. Remember to stay calm, take your time, and read the question and answers carefully before selecting an answer.

Notice that each line above contains the same access list number (1). This indicates that each line belongs to the same access-list, and that the access list is a Standard IP access list. When applying this access list, the router will compare the IP source address in each packet to the criteria specified by the "source" parameter in each line of the access list. Comparisons will be made to each line of the access list, one line at a time, in the exact order in which they appear above, from top to bottom, until a match is made or until the "implicit deny any" is reached. In the example above, the implicit deny any will never be reached, because all traffic that does not match any earlier lines will be caught by the "permit any" in the last line of the access list.

What does this access list do? First, it denies any traffic from the host with the IP address 10.10.10.10. This is accomplished with the 0.0.0.0 wildcard mask. Then it permits traffic from any host whose IP address contains the value 10 in the first three octets. All of the first three octets must contain the value 10 for this line to be considered a match, thanks to the 0.0.0.255 wildcard mask. Next, it denies any traffic from hosts whose IP addresses contain the value 10 in the first two octets. The values in the third and fourth octets will not matter, since we have a 0.0.255.255 wildcard mask. Finally, all other traffic will be permitted, thanks to the "permit any" line.

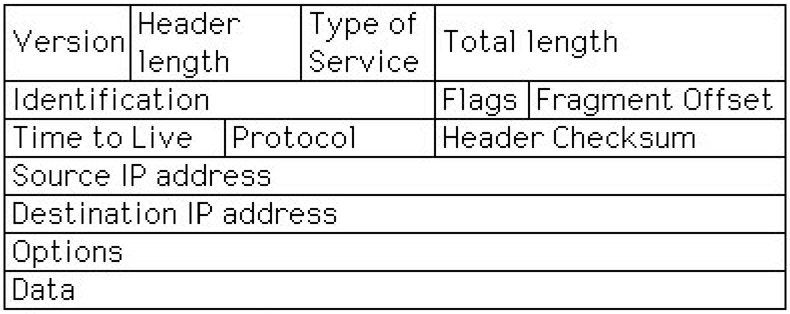
Notice that as the access list is executed from top to bottom, the comparison criteria become more and more general. This is the best way to build an effective access list. Since the order in which you enter your criteria is critical, put your most specific criteria in the access list first, and then broaden your scope with each successive line.

Once you have created an access list, there is no way to effectively edit it on the router, except to add additional lines to the end of the list. If this is not what you intend to do, you should copy the router configuration to a file and edit it with your favorite ASCII editor, then move it back to the router. Otherwise, you will need to completely re-write the access list from scratch to edit it. Before you begin to create an access list with a given number, or if you wish to completely re-write an existing access list, you should use the **no access-list** IOS command. For example, if you wish to create access list number 37, or if you wish to re-write access list 37 from scratch, you should first use the command:

```
no access-list 37
```

This will make sure that any previously configured lines for access list 37 are removed from the router configuration before you enter any new lines.

At this point in our discussion, it will be useful to briefly review some of the details of TCP/IP. This will not be a thorough discussion of the protocol suite, only a review of some of the points that are relevant to the topic of access lists. First, let's take a look at the IP packet format, which is illustrated in Figure 1.



Version	Header length	Type of Service	Total length	
Identification			Flags	Fragment Offset
Time to Live	Protocol		Header Checksum	
Source IP address				
Destination IP address				
Options				
Data				

[NA-SECU-WP1-011]

Figure 1: IP packet format

When considering the IP protocol, there are a few things to keep in mind. First, IP is a connectionless protocol. Second, in order for IP packets to be delivered correctly there must be a Source IP address and a Destination IP address declared within the packet. Finally, there should be some way to indicate the higher layer protocol that is to receive the IP packet's data. These objectives are accomplished by the Source IP address, Destination IP address, and Protocol field of the IP packet header.

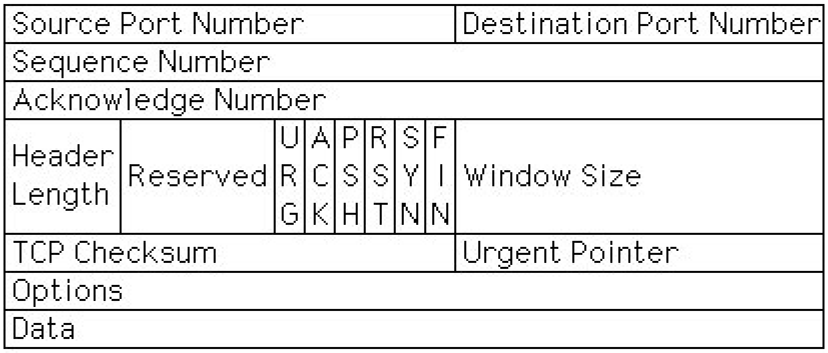
Extended IP Access Lists

So far, while using Standard IP access lists, we have only been concerned with the Source IP address field of the IP packet header when making the decision to permit or deny. A quick examination of the packet format above shows that there may be much more information of interest to us that would allow for much greater flexibility in our filtering. In order for us to make use of this information, we need to use Extended IP access lists. Extended IP access lists allow us to use the other fields in the packet header to make filtering decisions. In the case of IP packets, we would be interested in the Destination IP address field as well as the contents of the Protocol field.

It might be obvious why we find the Destination IP address field interesting. Just as we have been filtering packets so far based on their source, we might also want to filter them based on their destination. Why do we care what the contents of the Protocol field are? Extended IP access lists give us the ability to filter packets based on information contained in the layer 4 header as well. The contents of the Protocol field in the IP header will allow the router to determine what type of layer 4 header to expect.

Different higher layer protocols within the TCP/IP suite require different information to be specified within the header at layer 4.The information that will be interesting to us when creating access lists starts with whether the protocol is connection-oriented or connectionless. Connection-oriented protocols will use TCP as the layer 4 protocol.

Let's take a look at the TCP header:

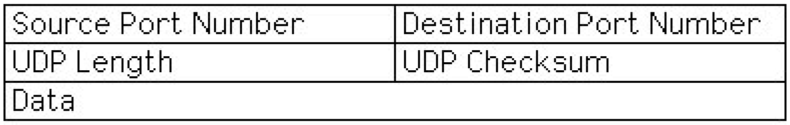


<table>
<tr><td colspan="8">Source Port Number</td><td>Destination Port Number</td></tr>
<tr><td colspan="9">Sequence Number</td></tr>
<tr><td colspan="9">Acknowledge Number</td></tr>
<tr><td>Header Length</td><td>Reserved</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window Size</td></tr>
<tr><td colspan="8">TCP Checksum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="9">Options</td></tr>
<tr><td colspan="9">Data</td></tr>
</table>

[NA-SECU-WP1-02t]

Figure 2: The TCP header

Some higher layer protocols within the TCP/IP suite use a connectionless layer 4 protocol called UDP. Here is the UDP header:



Source Port Number	Destination Port Number
UDP Length	UDP Checksum
Data	

[NA-SECU-WP1-03t]

Figure 3: The UDP header

In either case, the fields that concern us most are the Source Port number and the Destination Port number fields. Each of the many higher-layer protocols within the TCP/IP suite has a port number associated with it. This number is the mechanism that allows the transport layer to support multiple higher-layer protocols. The port number uniquely identifies the upper layer process that is the source or recipient of any given piece of data. Well-known port numbers are those numbers that are assigned by the Internet Assigned Numbers Authority (IANA), and are documented in RFC 1700.

Some of the common port numbers that you may see include:

FTP	TCP 21
TELNET	TCP 23
SMTP	TCP 25
DNS	TCP 53, UDP 53
TFTP	UDP 69
SNMP	UDP 161

Figure 4: Some Well-Known Port numbers

In order to filter IP traffic based on the higher layer protocol in use, you would specify the TCP or UDP port number associated with that protocol in your access list. Now that we have touched on the basics of ports, let's move on and examine the specific syntax of Extended IP access lists.

The syntax for Extended IP access lists extends the syntax you've already learned for Standard IP access lists. Initially, the syntax statement can be quite an eyeful, but after close examination it becomes easy to decipher. There are slight variations depending on whether the protocol uses TCP or UDP, so we'll cover them each separately. First we'll cover TCP. Take a look:

access-list *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} tcp *source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**established**] [**precedence** *precedence*] [**tos** *tos*] [**log**]

Confusing? No problem. Let's look at all of the new keywords and parameters and define them. Once you understand the purpose of each, this will no longer seem like such a mess. Keep in mind that not all of the keywords are appropriate all of the time. The lines in your actual Extended IP access lists will not get this complex very often. Here is a breakdown of the syntax statement above:

access-list: The IOS access-list command indicates that you are configuring a line in a particular access list.

access-list-number: The access list number uniquely identifies the access list on the router and also indicates its type.

dynamic dynamic-name [timeout minutes]: The optional keyword dynamic indicates that this access list is only valid for a limited time. The timeout parameter specifies the amount of time that an access list entry remains in a dynamic access list. Dynamic access lists are outside the scope of the CCNA exam, and consequently are outside the scope of this paper.

deny|permit: Specifies the action to take if the conditions in the access list line are met.

tcp: Specifies the protocol to examine. In this case, TCP is specified.

source: The IP address of the source station.

source-wildcard: The wildcard mask to apply to the source IP address for additional filtering.

operator: The operator field indicates which logical comparison you wish to make with the value for the source or destination port. Valid operators are:

```
lt - less than
gt - greater than
eq - equal to
neq - not equal to
range - an inclusive range
```

The range operator requires that you specify two port numbers; all port numbers between and including the values you specify will constitute a match.

The position of the operator and port number(s) indicates whether the match is associated with the source or destination port. If they immediately follow the *source* and *source-wildcard* keywords, then they must match the source port. If they immediately follow the *destination* and *destination-wildcard*, they must match the destination port.

port [port]: The port parameter indicates the port number that will constitute a match. In the case of a range, two port numbers must be specified -- a low and high value. The port parameter may be a number between 0 and 65535. For many of the well-known port values, this parameter may be a name instead of a number. Valid names include (but are not limited to) telnet, ftp, tftp, and domain.

Lock and Key

Though dynamic access lists are beyond the scope of this paper, here is a bit of information about them and a pointer to where you can find out more.

Dynamic access lists are a key feature in what Cisco calls Lock-and-Key Security. The basic purpose of Lock-and-Key Security is to provide access to a specific source/destination host through a user authentication process. This process involves the creation of appropriate access lists dynamically, as needed, and the removal of these access lists after a predetermined period of time.

Basically, this is how it works:

1. You initiate a Telnet session to the router.

2. You are authenticated. (This is a different authentication process than the simple vty password).

3. The router creates an entry in the dynamic access list.

4. You do whatever it is that you connected to the router to do and then end your session.

5. The access list entry is removed.

Cool, huh? For more detailed information regarding dynamic access lists, refer to the Lock-and-Key Security section of the documentation for your version of IOS. For version 11.2, it can be found online.

(is not associated with Cisco.)

destination: The IP address of the destination station.

destination-wildcard: The wildcard mask to apply to the destination IP address for additional filtering.

established: The established keyword causes a match to occur if the ACK or RST bits of the TCP segment are set. This would only occur if there has already been a session established. In the case of an initial session request, these bits would not be set.

precedence: Matches the value of the Precedence field of the IP header.

tos: Matches the value of the Type Of Service field of the IP header.

log: Indicates that a message should be sent to the console when a match occurs. The log message will be sent to the console after the first match of this line, and then a summary message will be sent at 5-minute intervals thereafter, indicating the number of packets that have matched this line in the previous 5 minutes.

Finding Ports

The Internet Assigned Numbers Authority (http://www.iana.org/) is the definitive source of port number assignments. The most commonly used port numbers are defined in http://www.isi.edu/in-notes/rfc1700.txt

In the formal port assignment process, 0-1023 are assigned to "well known," standards-based services. 1024-2047 may be registered voluntarily by vendors who wish to avoid conflict with other server port numbers. Port numbers above 2047 usually -- but do not always -- suggest client ports.

Now that you know what each keyword and parameter means, I am sure that you feel relieved. Extended IP access lists are easy once you become familiar with them. Let's examine a few specific examples to help you put all this information into practice.

Consider the following line:

```
access-list 101 permit tcp any any
```

This is perhaps the most basic form of an Extended IP access list. It is also probably useless in practice, as it allows all tcp traffic to flow through the router, but it is useful here to illustrate that the syntax can be fairly simple, depending on what you are trying to accomplish. Here is a more useful example:

```
access-list 101 permit tcp 200.199.198.0 0.0.0.255 any eq 23
```

The access list line above permits telnet traffic to any destination from any host on the 200.199.198.0 network. How? Like this -- the **access-list** command indicates that we are configuring an access list entry. The number of this particular access list is 101, indicating that we are configuring an Extended IP access list. The action to take when a packet matches this line is to permit the traffic. The protocol we are to examine is TCP.

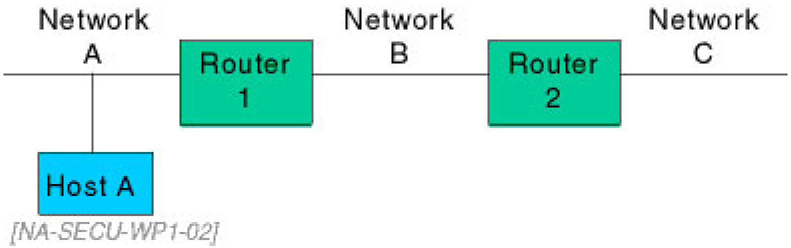
The source IP address and wildcard mask is 200.199.198.0 0.0.0.255. This indicates that in order to

Where to Filter

In addition to the syntax of the access list command lines, and the logical order of each access list entry, there is another issue to contend with when using access lists. This issue is placement. Once you have identified the traffic that you want to filter, you must consider the best location to place that filter.

Access lists are processor intensive for the router that executes them. You must make sure that the router you plan to use for filtering has the processor capacity to handle the increased load that the access lists will place on it.

Deciding where to place your access lists will be determined in large part by the purpose of the access list. Consider this small scenario:



Let's say you simply want to prevent Host A from reaching Network C. You create a simple Standard Access list to filter traffic from Host A. Where do you place this access list? You must place it on Router 2. If you place the access list on Router 1, then Host A will not only be unable to reach Network C, but it will also be prevented from reaching Network B. This is probably not what you intended. You might place the access list on Router 2 as an outbound access list on the interface directly connected to Network C, and this would work. It might be better, however, to place the access list on Router 2 as an incoming access list on the interface directly connected to Network B. Why? This will deny the traffic before it enters Router 2. This reduces the impact of the traffic on Router 2 by dropping it before it goes through the routing process. It makes little sense in this scenario to allow the traffic to be routed only to drop it before it leaves the router.

With Standard IP access lists, placement is somewhat easier to determine due to the fact that they can only filter traffic based on source address. Give your scenario a bit of thought, and it becomes easy to spot issues like those we identified in the simple example above. Standard IP access lists usually must be placed as close as possible to the destination in order to filter the traffic where it is unwanted, but allow that traffic to reach other points within the internetwork.

Extended IP access lists offer greater flexibility in filtering, but this flexibility can make placement decisions less obvious. Sometimes access list placement issues become a trade off between network bandwidth and ease of administration. In general, placing Extended IP access lists close to the source of the traffic is a good idea in order to keep traffic that will ultimately be dropped from travelling too far through the internetwork and using up valuable bandwidth. This also helps to reduce the impact of ICMP messages that might get sent to the source in response to the traffic getting dropped. These messages include Host Unreachable and Network Unreachable messages.

On the other hand, it can simplify administration to place access lists closer to the core of your network rather than distribute them out to the edge of the network. Locating the access lists closer to the core might enable you to consolidate them, giving you fewer lists to maintain. This becomes an even more important consideration as your internetwork grows.

Access Lists' Impact on Performance

As mentioned earlier in this paper, access lists can be very CPU intensive for the router that executes them. In addition, access lists have an impact on a router's switching function. Autonomous switching cannot be performed on any interface that uses Extended IP access lists. Likewise, silicon switching cannot be performed on interfaces configured to use dynamic access lists. These issues need to be considered when you decide whether to use access lists and where to place them.

Monitoring Access Lists

It is often useful to examine the contents of the access lists on your router. To do this, use the IOS command **show access-list**. This command, by itself, will display the contents of all access lists that are configured on the router, regardless of type, name, or number. To view only IP access lists configured on the router, use the **show ip access-list** command.

To view a specific IP access list, use the access list number with the **show ip access-list** command. For example, to view only the contents of access list 101, use the following command:

```
Router#show ip access-list 101
```

Conclusion

If you have read this entire paper and absorbed the material presented within it, you know now all that you will need to know to correctly answer the questions on Network Security that you will find on the CCNA exam. You may even feel comfortable setting up access lists on your own routers. Your comfort will increase as you use them and become more familiar with their application, their impact, and their performance.

You will be asked to solve Network Security problems on the CCNA written exam. Are you ready to solve them?

References

Cheswick & Bellovin (1994). *Firewalls and Internet Security: Foiling the Wily Hacker.* Reading, MA: Addison-Wesley-Longman.

Chapman & Zwicky (1996). *Building Internet Firewalls.* Sebastopol, CA: O'Reilly.

RFC 2196. Site Security Handbook.

Stevens (1994). *TCP/IP Illustrated, Volume 1 The Protocols*. Reading, MA: Addison-Wesley-Longman.

Chappell (1999). *Advanced Cisco Router Configuration.* Indianapolis, Indiana: Cisco Press, Macmillan Technical Publishing.