7
CHAPTER

NETWORK SECURITY

Chapter Outline

Objectives

- Review denial of service attacks
- Introduce the procedures for configuring access lists
- Examine "best practice" for router security
- Examine "best practices" for switch security
- Examine the issues of wireless security
- Introduce the steps for configuring VPNs

Key Terms

denial of service (DoS)
SYN attack
smurf attack
spoof
directed broadcast
hacked
distributed denial of service (DDoS) attack
firewall
access lists (ACL)
packet filtering
proxy server
stateful firewall
demilitarized zones
access lists
SMB
edge router
permit ip any any
host
filter list
line password
EXEC level password
Type 7
Type 5
AAA
transport input none
crypto key generate rsa
CDP
NTP
switchport port-security
violation action
protected
restrict
shutdown
ERRDISABLE
storm control
pps
rising threshold/falling threshold
STP Portfast
BPDU guard
BPDU filter
STP Root guard
DTP
SSID
Beacon
open authentication
Sharekey Authentication
WEP
WPA
TKIP
AES
CCMP
LEAP
EAP
RADIUS
VPN
IP tunnel
GRE
PPP
PAP
CHAP
EAP
MD5
RADIUS
PPTP
L2F
L2TP
AH
ESP
SHA-1
DES, 3DES
AES
IKE
ISAKMP
Diffee-Hellman

INTRODUCTION

This chapter examines the topics of network security. The concept of denial of service (DoS) is first examined in Section 7-1. This section also examines the SYN attack, smurf attack, and distributed denial of service (DDoS) attacks. Section 7-2 examines firewalls and access lists. Topics included in the section include stateful firewalls, demilitarized zones, and configuring access lists. Sections 7-3 through 7-5 look at "best practices" for setting up security on routers, switches, and wireless networks. This chapter concludes with a look at configuring external access to networks using a VPN in Section 7-6.

7-1 DENIAL OF SERVICE

Denial of Service (DoS)
Means that a service is being denied.

Denial of service (DoS) means that a service is being denied to a computer, network, or network server. DoS attacks can be on individual machines, on the network that connects the machines, or on all machines simultaneously.

You can initiate a DoS attack by exploiting software vulnerabilities. For example, a software vulnerability can permit a buffer overflow, causing the machine to crash. This affects all applications, even secure applications. A database of software vulnerabilities is available online at http://web.nvd.nist.gov/.

The vulnerable software DoS attack attacks the system by making it reboot repeatedly. DoS attacks can also be on routers via the software options that are available for connecting to a router. For example, SNMP management software is marketed by many companies and is supported by many computer platforms. Many of the SNMP packages use a similar core code that could contain the same vulnerability.

SYN Attack
This attack refers to the opening up many TCP sessions to limit access to network services.

Another DoS attack is a **SYN attack**. This refers to the TCP SYN (synchronizing) packet. An attacker sends many TCP SYN packets to a host, opening up many TCP sessions. The host machine has limited memory set aside for open connections. If all the TCP connections are opened by the SYN attack, other users are kept from accessing services from the computer, because the connection buffer is full. Most current operating systems take countermeasures against the SYN attack.

Smurf Attack
A way of generating a large amount of data traffic.

Spoof
Inserting a different IP address in place of an IP packet's source address to make it appear that the packet came from another network.

DoS attacks can affect the network bandwidth and the end points on the network. The classic example is the **smurf attack** (see Figure 7-1), which requires few resources from the attacker.

In the smurf attack shown in Figure 7-1, the attacker sends a small packet and receives many packets in return. The attacker then picks a victim and an intermediate site. Figure 7-1 shows an attacker site, an intermediate site, and a victim site. The intermediate site has subnets of 10.10.1.0 and 10.10.2.0. The victim is at 10.10.1.0. The attackers send a packet to 10.10.1.255, which is a broadcast address for the 10.10.1.0 subnet. The attacker will **spoof** the source address information, making it look as if the packet came from the victim's network. *Spoof* means the attacker doesn't use his IP address but will insert an IP address from the victim's network or another network as the source IP. All the machines on the 10.10.1.0 subnet will send a reply back to the source address. Remember that the attacker has spoofed the

source address so the replies will be sent to the victim's network. If this attack were increased to all the subnets in the 10.0.0.0 network, an enormous amount of data packets will be sent to the victim's network. This enables the attacker to generate a lot of data traffic on the victim's network without requiring the attacker to have many resources.

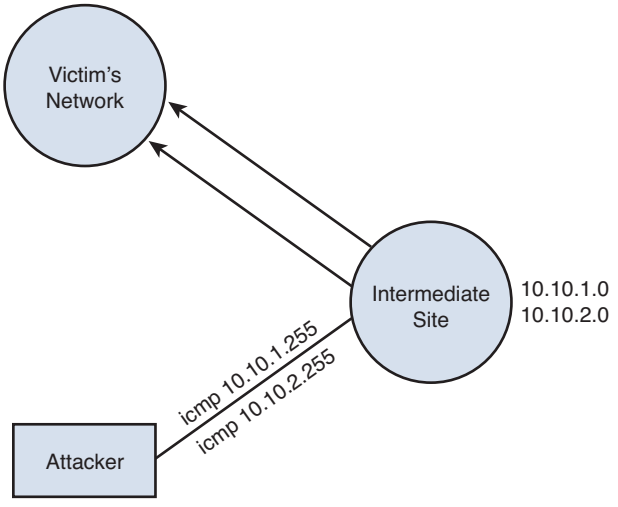

FIGURE 7-1 An example of a smurf attack

This type of attack is not new, and there are some steps that can be taken to stop a network from becoming an intermediate site. Cisco routers have an interface command that blocks broadcast packets to that subnet. This prevents a network from becoming an intermediate site for a network attack such as this. Make sure this command or a similar command is a default or has been enabled on the router's interface:

```
no ip directed-broadcast
```

But aren't Layer 3 devices supposed to stop broadcasts? This is true for general broadcasts (all 32 bits set to 1s or F F F F F F F F or 255.255.255.255). Routers will always stop these broadcasts. The type of broadcast used in a smurf attack is a **directed broadcast**, which is passed through the router. The **no ip directed-broadcast** command enables only the router to reply.

Directed Broadcast
The broadcast is sent to a specific subnet.

To prevent your network from becoming a host for an attacker, use access lists to allow only specific sources from the network to enter the router's interfaces. For example, network B connects to a router. Only packets sourced from network B are allowed to pass through the router. The downside of this is it does become a maintenance problem: Keeping track of the access lists can be a challenge for the network administrator and processing access lists on the router is processor intensive and can slow the throughput of the packets. Access lists do help eliminate spoofed packets, however. There is a lot of software on the Internet that enables someone to spoof an IP address. To prevent yourself from becoming a victim, well…there isn't a way unless you aren't connected to any network or to any other users.

Distributed Denial of Service Attacks (DDoS)

Hacked
Attacked.

Distributed Denial of Service (DDoS) Attack
An attack that comes from a potentially large number of machines.

The number of packets that can be generated by a single packet as in the smurf attack can be limited on a router; however, attackers now use worms to distribute an attack. The attacker will do a port scan and look for an open port or a software application that is vulnerable to an attack. The machine is ***hacked*** (attacked) and distributes the malicious software. The attacker will repeat this for many victim machines. Once the software is on the victim machines, the attacker can issue a command or instruction that starts the attack on a specific site. The attack will come from a potentially massive amount of machines that the worm has infected. This is called a **distributed denial of service (DDoS) attack.** To stop DDoS attacks, stop intrusions to the network. The bottom line is **PREVENT INTRUSIONS**.

7-2 FIREWALLS AND ACCESS LISTS

Firewall
Used in computer networks for protecting the network.

Access Lists (ACL)
A basic form of firewall protection.

Firewalls are used in computer networks for protection against the "network elements" (for example, intrusions, DoS attacks, etc.). **Access lists (ACLs)** are the basic form of firewall protection, although an access list is not stateful and is not by itself a firewall. Access lists can be configured on a router, on a true dedicated firewall, or on the host computer. Firewalls are examined first in this section.

Firewalls allow traffic from inside the network to exit but don't allow general traffic from the outside to enter the network. The firewall monitors the data traffic and recognizes where packets are coming from. The firewall will allow packets from the outside to enter the network if they match a request from within the network. The followings are some of the well-known technologies that most firewalls are based on:

- Packet filtering
- Proxy server
- Stateful packet filtering

Packet Filtering
A technique used to determine whether a packet is allowed to enter or exit the network.

Packet filtering is a technique used to determine whether a packet is allowed to enter or exit the network based on its Layer 3 IP header information, such as source IP address and destination IP address, or its Layer 4 header information, such as protocol and port number. A limit is placed on the packets that can enter the network or it can be used to limit information moving from one segment to another. ACLs are used to enable the firewall to accept or deny data packets. The disadvantages of packet filtering are as follows:

- Packets can still enter the network by spoofing or fragmenting the data packets.
- It is difficult to implement complex ACLs.
- Not all network services can be filtered.

Proxy Server
An agent for handling requests from clients seeking resources.

A **proxy server** is used by clients to communicate with secure systems using a proxy. Essentially, the proxy server acts as an agent for handling requests from clients seeking resources, such as access to the network. This step is used to authenticate the user, establish the session, and set policies. The client must connect to the

proxy server to connect to resources outside the network. The disadvantages of the proxy server are as follows:

- The proxy server requires processing power.
- Adding services can be difficult.
- There can be a potential problem with network failure if the proxy server fails or is corrupted.

In a **stateful firewall**, the state of inbound and outbound data packets are tracked and compared to determine if a connection should be allowed. This includes tracking the source and destination port numbers and sequence numbers, as well as the source and destination IP addresses. Stateful packet filtering, sometimes referred to as dynamic packet filtering, monitors the session or state of the connection initiated from the trusted network and allows the return traffic to enter the network. This technique is used to protect the inside of the network from the outside world but still allow traffic to go from the inside to the outside and back. The firewall needs to be stateful to accomplish this.

Stateful Firewall
The inbound and outbound data packets are compared to determine if a connection should be allowed.

For example, a machine called NVL is on the inside of a network. NVL establishes a connection to the outside at www.network-A.edu. The connection requires the initial TCP handshake sequence and the first SYN packet hits the firewall. The firewall has been configured to allow packets to leave the network. The firewall recognizes that a connection is being established outside the network and the firewall creates a state that includes the source and destination IP address numbers for the connection. The TCP packets arrive at www.network-A.edu (port 80) and the server at networkA.edu returns a SYN-ACK packet back through the firewall. The firewall examines the SYN-ACK packet, and matches the stored source and destination IP addresses with the packet's source/destination IP addresses and port numbers. If the information matches, the IP packets are allowed to pass. This repeats until the connection ends.

What if an attacker tries to spoof the firewall to gain access to the interior of the network?

In this case, a connection already exists between NVL and www.network-A.edu. The attacker spoofs the network-A.edu domain www server's IP address and port 80 (the web server) and tries to use this to gain access to the network. Remember that there is a sequence number associated with the data transfers in the TCP connection. The server recognizes that there is a discrepancy with the sequence and rejects the hacker's connection, preventing the attack.

But, what if the campus network has a web server? How are outside users allowed access?

This requires that holes must be opened in the network that allow data packets to pass through. The three most common traffic types that require holes to be opened are web servers, DNS, and email. The firewall must be modified so that anybody can connect to the web server via port 80. But, what if a vulnerability is discovered on port 80 for the server's operating system? When you open ports, the network administrator must continually upgrade the software so that vulnerabilities are removed. The web server may also need to have its own firewall. Most firewalls can perform deep packet inspection. This may catch some of the protocol vulnerabilities.

A big problem with firewalls is that users assume a firewall catches all possible problems. This is a wrong assumption. The user may be slow to update the patches and fixes to the software. For example, an attacker sends an email message with an attachment to a user. The user opens the attachment and unknowingly loads a Trojan horse on his or her computer that scans all of the machines on the LAN, checking for any possible open ports, compromising the entire LAN. A firewall is not the end-to-end solution.

Network Attack Prevention

Demilitarized Zones
A physical or logical network designed to house the public servers that will have direct exposure from the outside network.

A general rule of thumb is to place the firewall close to the machines you want to protect (for example, the network servers). Do not assume that the clients on the network will never attack your system. Create **demilitarized zones** (DMZ) for the public or outside servers, which mean that they are moved to a segment on the network so that they are separated from the inside or trusted network. The DMZ is a physical or logical network designed to have the contact with and the exposure from the outside or untrusted network. This limits the direct exposure of the inside network from the outside network. If the machines are compromised, the intruder will have limited access to the inside of the network. Keep in mind that firewalls do not protect the network from viruses or malwares. Clients can and will get viruses on their machines. Other countermeasures such as antivirus policies and OS patches must be deployed along with firewall for more effective security.

Firewalls are not the solution for everybody. Open networks, such as a university's, have limited areas where a firewall can be placed. For example, firewalls will be placed close to critical machines, such as academic records. There are so many entities on a university campus network that need connections around the world. The university campus network will have multiple web servers that can't be centrally located. If a firewall was placed on the whole university network, many holes would be required and thus negate the usefulness of the firewall. One solution is to put in server firewalls.

Access Lists

Access Lists
Used to tell a networking device who and what are allowed to enter or exit a network.

Access lists provide basic protection for the network. It is what tells a networking device who and what are allowed to enter or exit a network. The access list compares the source and destination IP address and the source and destination port numbers and sometimes might examine the packet contents above Layer 4 (transport). However, access lists primarily focus on the network (Layer 3) and transport (Layer 4) layers. A router is often placed on the edge of a network to handle data traffic entering and exiting the network, and it is common practice to block some data traffic. The first two steps for applying access lists on a router are

1. Identify the problem.
2. Decide where to place the access list.

There can be many problems encountered on a network that require the application of an access list to a router. For example, the network administrator will block certain types of data packets from entering and exiting a network. For example, the

Microsoft NetBIOS protocol for mapping drives (also called **SMB** [server message block] over TCP) is an intranet protocol and is not intended to be run over the Internet. SMB data packets use ports 137, 138, and 139. SMB packets will be blocked from entering and exiting the network. The next issue is where to place the access list.

In this case, the best place to apply the access list is the network's Internet connection. The following discussion describes the steps for applying an access list to a router. The network management protocol SNMP (port 161) can be blocked to prevent an outside attacker from getting into your router(s). The following is an example of how to configure an **edge router** to block SNMP from entering a specific LAN.

In this case, the term "edge router" is describing the Internet connection to the campus network, and the LAN being protected is LAN B. The network topology is shown in Figure 7-2.

SMB
Server message block. A protocol used by Windows computers to share folders, printers, and serial ports to other computers on the same network.

Edge Router
Describes the Internet connection to network.

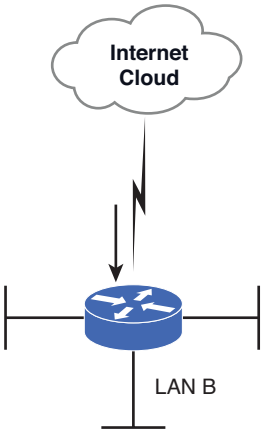

FIGURE 7-2 An example of setting an access list on an edge router to block SNMP data packets

The first step for configuring the access list is to enter the router's configuration mode using the **configure terminal** command, as shown:

```
RouterB# conf t
RouterB(config)#
```

Next, define the access list to be applied to the router interface. Access lists can be specified in two ways: They can be either a standard or extended type. A *standard* access list is used when specifying access for only IP addresses. An *extended* access list allows the addition of port numbers. For example, the access list could be defined to deny SMB data packets to enter or exit the network. In the following example, an extended access list of 100 is being defined with the instructions to deny UDP packets from any source going to any destination equal to port 161 (SNMP). (*Note:* The 100 is just an identifier used to indicate what list is being defined.) The command **access-list 100 deny udp any any eq 161** is entered.

This command is used to deny all SNMP data packets from entering or exiting the network. Remember that the first step is to identify the problem. Unauthorized access to the network's router must be prevented, so the access list is being applied. SNMP uses the UDP protocol for transferring data; therefore, it is also a good idea to block UDP packets. The command **access-list 100 deny udp any any eq snmp** is used to instruct the router to deny UDP packets from any source to any destination equal to SNMP. In this example, SNMP is used instead of 161. Cisco routers allow the use of names for well-known port numbers, for example, SNMP (port 161). The entry of these two commands from the router's (config)# prompt is provided:

```
RouterB(config)# access-list 100 deny tcp any any eq 161
RouterB(config)# access-list 100 deny udp any any eq snmp
```

permit ip any any
The instruction added to the last line of an access list to allow all other data packets to enter and exit the router.

These commands form an access list that blocks TCP and UDP data packets from any source going to any destination equal to SNMP (port 161). Even though SNMP uses UDP port 161 as defined by the RFC, it is not uncommon to see network administrators blocking the same port number for both TCP and UDP as a precaution. There is an implicit denial at the end of an access list in Cisco routers, and this statement alone will block all data packets. The access lists must be modified to permit any other data packets to enter and exit the LAN. The command **access-list permit ip any any** must be added to the last line of an access list to allow all other data packets to enter and exit the router. An example is shown here. This instruction is for access-list 100, and it instructs the router to permit IP packets from any source to any destination:

```
RouterB(config)# access-list 100 permit ip any any
```

The content of the access list just created can be checked using the command **show access-list 100**, as demonstrated here:

```
RouterB# sh access-list 100
Extended IP access list 100
deny tcp any any eq 161
deny udp any any eq snmp
permit ip any any
```

The next step is to decide where to place the access list. Specifically, this is asking what router interface is to be used to apply the list. In this case, the access list is to be applied to the Serial0/0 interface and to the inbound data packets (coming from the Internet). The access list can be applied to both in and outbound data packets. The format of the command is

```
Router (config)# int s0/0
Router (config-if)# ip access-group 100 in
```

The **100** matches the number from the access list being applied. This access list is being applied on the *in direction.* This denies any SNMP packets from the outside into the interface serial0/0. If this command was modified to **ip access-group 100 out**, this would deny any SNMP packets coming from inside the network going out. This would prevent users on the network to do SNMP queries on the Internet, but it would allow SNMP queries from the outside to come in. This is obviously the opposite of the intent of the access list. There is a limit on applying access lists to a

router's interface, one access list–in per interface and one access list–out per interface. The following example demonstrates the use of an access list.

Example 7-1

Problem: Develop an access list to prevent any port 137 SMB data packets from anywhere or going anywhere to enter or exit the router's Serial0/0 interface.

Solution:

Configure the router to use access list 120:

```
access-list 120 deny tcp any any eq 137
access-list 120 permit ip any any
```

Apply the access list to the Serial0/0 interface:

```
ip access-group 120 in
ip access-group 120 out
```

Extended access lists identifiers are not restricted to numbers. A more convenient way to identify them is to use a name to describe the purpose of the access list. For example, in the previous example, an access list called block-snmp could be used. The network administrator can quickly identify the purpose of the list without having to check each entry. An example is shown:

```
LAN-B(config)#ip access-list extended block-snmp
LAN-B(config-ext-nacl)#deny tcp any any eq 161
LAN-B(config-ext-nacl)#deny udp any any eq snmp
LAN-B(config-ext-nacl)#permit ip any any
LAN-B(config-ext-nacl)# end
LAN-B#sh access-list
Extended IP access list 100
10 deny tcp any any eq 161
20 deny udp any any eq snmp
30 permit ip any any
Extended IP access list block-snmp
10 deny tcp any any eq 161
20 deny udp any any eq snmp
30 permit ip any any
```

Notice the **show access-list** command lists both the 100 and the block-snmp access lists. Both lists actually do the same thing. Next, the block-snmp access list is applied to the router's Serial0/0 interface, as shown:

```
LAN-B(config)#int s0/0
LAN-B(config-if)#ip access-group block-snmp in
```

Example 7-2 demonstrates an application of the named access list.

Example 7-2

Problem: Create an access list to block any UDP packets from the network 10.66.66.0/24 to enter the Serial0/1 interface on a router. Specify an extended access list of block-udp.

Solution:

From the router's (config)# prompt:

```
ip access-list extended block-udp
deny udp 10.66.66.0 0.0.0.255 any
permit ip any any
```

On the serial interface:

```
interface s0/1
ip access-group block-udp in
```

Notice that the access list uses the wild card mask 0.0.0.255 to describe the network 10.66.66.0. The 0 indicates a "must match" and 255 indicates a "don't care." This results in a match for 10.66.66.0. An interesting statistic to look at is how many times an access list has been matched. The command **show access-list** followed by the name or number of the list enables the statistics to be viewed. The **(# matches)** indicates how many times there has been a match to the access list. An example is shown here for the blocksnmp access list:

```
LAN-B#sh access-list block-snmp
Extended IP access list block-snmp
deny tcp any any eq 161
deny udp any any eq snmp
permit ip any any (7 matches)
LAN-B#sh access-list block-snmp
Extended IP access list block-snmp
deny tcp any any eq 161 (6 matches)
deny udp any any eq snmp
permit ip any any (7 matches)
```

The first display for the **show access-list** indicates that no packets have been denied, but there have been seven matches for **permit ip any any**. The second group shows that TCP data packets have been denied six times. This information can be useful for the network administrator when evaluating the effectiveness of an access list. There are times when a host with a specific IP address needs to be denied. For example, an attacker could have gained access to a host computer in the network and configured it to continually ping a server, attempting to generate a denial of service or just to slow down data traffic.

Another possibility is the computer could be on a host external to the network that is continually pinging a server within the network. The network administrator examines the data traffic and determines that the IP address of the attack is from a remote computer (192.168.12.5). The computer is not in the administrator's LAN;

therefore, the data traffic must be stopped in the administrator's network. Configuring an access list to deny any packets from the remote host can do this. Assume that the data traffic is entering the network via the Serial0/0 interface on RouterB, the Internet router, as shown in Figure 7-3.

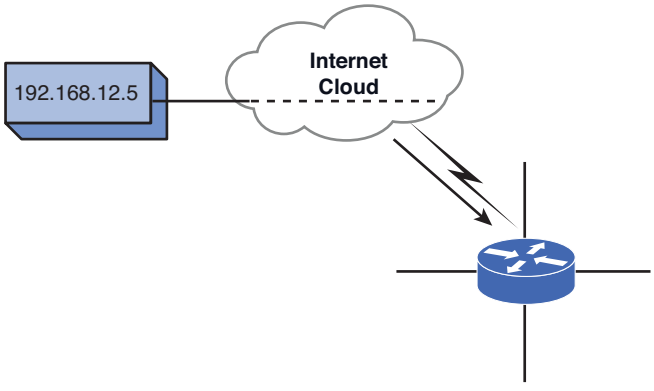

FIGURE 7-3 An example of applying an access list to stop data traffic from a remote host

```
The following demonstrates how the access list could be
configured:LAN-B(config)#ip access-list extended block-badguy
LAN-B(config-ext-nacl)#deny ip host 192.168.12.5 any
LAN-B(config-ext-nacl)#permit ip any any
LAN-B(config-ext-nacl)#int ser0/0
LAN-B(config-if)#ip access-group block-badguy in
```

This example shows that an extended access list called block-badguy was created. The access list denies source packets from the host 192.168.12.5 with any destination. The **host** entry enables a specific IP address to be entered. Next the access list is applied to the router's Serial0 interface in the in direction. The **in** describes that the list is applied to any data packets coming into the interface.

Host
Enables a specific host IP address to be entered in the access list.

Example 7-3 shows the blocking of a host IP address.

Example 7-3

Problem: Create an access list to block any data traffic from the IP address 192.168.12.5 to anywhere. Specify an extended access list name of border-router. Apply the access list to the router's Serial1 interface. Assume that the router interface is connected to the Internet, and the 192.168.12.5 host is external to the network.

Solution:

```
ip access-list extended border-router
deny ip host 192.168.12.5 any
permit ip any any
int s1
ip access-group border-router in
```

A bad thing about access lists is that they are not stateful. This means they don't keep track of the data packet flow. They only examine the current packet, and they contain no information about the recent history of data traffic on that connection. This makes it hard for access lists to allow connections to be made to the outside world. Access lists try to address this need for allowing a connection from the outside to the inside if an Established connection exists. This requires using a flag in the TCP header, but TCP headers can be falsified; therefore, the access list is useless for verifying an established connection. Someone could fake an acknowledgement flag so his or her packet could enter the network.

Filter List
Juniper's non-stateful packet filter.

Juniper Filter List The Juniper JUNOS has something similar to Cisco's access list called a **filter list**. Juniper's filter list is a non-stateful packet filter. A filter list is configured in a similar fashion as configuring a routing policy. The programming syntax in most part is the same. Filter lists are defined under the firewall section. Typically, filter lists are used in conjunction with prefix lists, which are groups of networks and IP addresses. This is for ease of programming and organizing.

The example that follows demonstrates configuration steps of how to create a JUNOS filter list. The following is a filter list to allow SNMP access for a trusted network of 192.168.10.0/24 and a trusted host of 10.20.20.32. This filter list will deny all SNMP traffic from everywhere else.

```
net-admin@noc# configure
Entering configuration mode
[edit]
net-admin@noc# set policy-options prefix-list snmp-list
192.168.10.0/24
[edit]
net-admin@noc# set policy-options prefix-list snmp-list 10.20.20.32/32
[edit]
net-admin@noc# edit firewall family inet
[edit firewall family inet]
net-admin@noc# # edit filter protect
[edit firewall family inet filter protect]
net-admin@noc# # edit term allow-snmp
[edit firewall family inet filter protect]
net-admin@noc# set term allow-snmp from source-prefix-list snmp-list
[edit firewall family inet filter protect]
net-admin@noc# set term allow-snmp from protocol udp
[edit firewall family inet filter protect]
net-admin@noc# set term allow-snmp from destination-port 161
[edit firewall family inet filter protect]
net-admin@noc# set term allow-snmp then accept
[edit firewall family inet filter protect]
net-admin@noc# set term block-snmp from protocol udp
[edit firewall family inet filter protect]
net-admin@noc# set term block-snmp from destination-port 161
[edit firewall family inet filter protect]
net-admin@noc# set term block-snmp then reject
[edit firewall family inet filter protect]
net-admin@noc# set term everything-else then accept
[edit firewall family inet filter protect]
```

A prefix list called snmp-list is created and it contains two trusted SNMP sources. A filter called protect has three terms: **allow-snmp**, **block-snmp**, and **everything-else**. The term **allow-snmp** permits the traffic from the prefix list snmp-list to UDP port 161 (SNMP). The term **block-snmp** denies any traffic to UDP port 161 and the term **everything-else** permits the rest of the traffic that neither match the term allow-snmp nor block-snmp to pass through. After the filter list is created, it can be applied as the input filter or the output filter to an interface similar to Cisco access-list. For example, to apply the preceding filter list to the interface ge-0/2/0 of a Juniper router, use the following command:

```
[edit]
net-admin@noc# set interface ge0/2/0 unit 0 family inet filter input
protect
```

You can verify the filter list configuration with the **show firewall** command, which will show any configuration within the firewall section:

```
[edit ]
net-admin@noc# # show firewall
family inet {
      filter protect {
           term allow-snmp {
            from {
            source-prefix-list {
                  snmp-list;
            }
            protocol udp;
            destination-port snmp;
            }
            then accept;
      }
      term block-snmp {
            from {
            protocol udp;
            destination-port snmp;
            }
            then reject;
      }
term everything-else {
      then accept;
      }
}
```

7-3 ROUTER SECURITY

Routers perform essential services for the network. The majority of the routers are deployed at the perimeter of the network. Therefore, they are the first line of defense of the network. Compromise of a router can lead to many issues on the network, such as degrading network performance, denial of network services, exposure of network configuration details and exposure of the sensitive data. A poorly

configured router can easily become a compromised router, thereby reducing the overall security of the network and potentially exposing the internal network to scans and attacks. The following section focuses on the "best practice" on how to configure a network router to avoid or prevent very serious security problems.

Physical security is always on top of the list of any best security practice. Routers should be placed in a secure area where it is accessible only to authorized personnel. Recall that the easiest access to a router is via its console port. If someone gains access to the premise, then they have physical control of the router. The router may be secured with a password, but the router's password can be recovered if someone has console access. Even worse, the router can become disabled or damaged and all network services will be halted until the situation is repaired.

The operating system of the router is another crucial component that any network administrator must keep up to date. However, it has been known that the latest version of any IOS is not the greatest. Its reliability is questionable because of its limited exposure of testing. Most network administrators will wait before upgrading to the latest version to make sure that there are no side effects or bugs. Most network administrators will settle for the latest stable release of the IOS, but not the very latest one.

Configuration hardening is needed to limit the exposure of a router. This section focuses on configuration hardening of Cisco routers. The same concepts still apply to other vendor routers; however, the configuration method and commands will be different.

Router Access

Line Password
Used to gain access to the router.

EXEC Level Password
Used to gain access to EXEC commands.

Type 7
Uses a Cisco encryption algorithm.

Type 5
Uses MD5 hash for encryption.

Local access and remote access to the router are the common ways of gaining control of a router, and access must be restricted to only authorized personnel. A typical way of securing the local access or remote access is to create a password. There are two types of passwords used on a router: the **line password** and EXEC password. The line password is used to gain access to the router and the privileged EXEC level password that is used to gain access to EXEC commands. The line password is recommended to be used in conjunction with the command **service-password encryption**. This global command will encrypt the password and display it in the encrypted form. It is not a strong encryption, but it can be used to provide low level security. The **EXEC level password** used to be enabled with the **enable password** command, but that command has been replaced by the **enable secret** command. This command provides a stronger password encryption. There are two password protection schemes used in Cisco IOS: Type 7 and Type 5. **Type 7** uses a Cisco encryption algorithm, which is not as strong as **Type 5** protection, which uses MD5 hash. Therefore, it is recommended that you use Type 5 protection whenever possible.

A security step beyond typical password protection is to create a user account for authorized personnel. This provides the capability to track and log each time a system is accessed. A local user account can be created on a router by using the command **username** [*name*] **privilege** [*level*] **password** *[password_string]* as shown:

```
RouterA(config)# username admin privilege 10 password @dm1np@$$wd
```

Cisco provides 16 levels (0–15) of privileges. Each level is pre-assigned with commands that can be run. Level 15 is the highest and is equivalent to privileged EXEC

mode. The command **username admin privilege 10 password @dm1np@$$wd** creates a local user called admin with privilege level 10. The drawback of creating a local user is that the same user has to be created on every router on the network. This does not provide scalability. Cisco offers Authentication, Authorization, and Accounting (**AAA**) service as a way to centrally manage and control user access. With AAA, two of the most used access protocols, RADIUS (Remote Authentication Dial In User Service) and TACACS+ (Terminal Access Controller Access-Control System Plus), are supported. This means that Cisco routers can communicate with RADIUS or TACACS+ servers for central authentication. AAA enables authentication based on the router's local user database, enable, line passwords, as well as other access protocols. The following shows an example of how to configure AAA on a Cisco router:

AAA
Authentication, Authorization, and Accounting.

```
RouterA(config)# aaa new-model
RouterA(config)# aaa authentication login default local group tacacs+
RouterA(config)# aaa authorization exec default local group tacacs+
if-authenticated
```

Once the authentication method is defined, it can be applied to any access entry point whether it is local or remote. The local access can be via the console port or the auxiliary port. The remote access is via VTY or virtual terminal. The following example shows how to configure a console port with security access. It enforces the authentication using the local user database and the timeout of 5 minutes if the user input is not detected. Also, it prevents the remote access to the console port via reverse-telnet with the command **transport input none**:

transport input none
This command prevents the remote access to the console port via reverse-telnet with the command.

```
RouterA(config)#line con 0
RouterA(config-line)#login local
RouterA(config-line)#exec-time 5 0
RouterA(config-line)#transport input none
```

The remote access to the router can be made via Telnet or SSH. Telnet is a default transport protocol into a router, but its unencrypted traffic is a big security flaw. Therefore, SSH is recommended whenever possible. To enable the SSH transport requires an extra step of generating an RSA key. To generate an RSA key, the hostname and the domain name must be pre-configured on the router as this information will be used as part of the key. To generate the key, the command **crypto key generate rsa** is issued:

crypto key generate rsa
Command used to generate an RSA key.

```
RouterA(config)#crypto key generate rsa
The name for the keys will be: RouterA.et477.local
Choose the size of the key modulus in the range of 360 to 2048 for
your
  General Purpose Keys. Choosing a key modulus greater than 512 may
take a few minutes.
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

After the RSA key is generated, the remote VTY access can be configured with SSH as the transport. The following is an example configuring the VTY remote access with SSH. The configuration uses the default login authentication that was defined in the AAA section. The command **transport input ssh** enforces SSH as the

only access method. The access-class 15 defines the access-list of the network that is allowed to connect to the router via SSH. Lastly, the exec-timeout of 5 minutes is configured:

```
RouterB(config)# access-list 15 permit 10.10.20.0 0.0.0.255
RouterB(config)# access-list 15 deny any
RouterA(config)# line vty 0 4
RouterA(config-line)# access-class 15 in
RouterA(config-line)# login authentication default
RouterA(config-line)# transport input ssh
RouterA(config-line)#exec-time 5 0
```

Router Services

A router has many services enabled by default. These services vary from vendor to vendor. Unnecessary services should be disabled and those services deemed necessary should be tightened. The TCP/IP services like echo, discard, daytime, chargen, bootps, finger, identd, and snmp, are enabled automatically on a Cisco router and most of these are not needed. They can be disabled globally as follows:

```
RouterA(config)#no service tcp-small-servers
RouterA(config)#no service udp-small-servers
RouterA(config)#no ip bootp server
RouterA(config)#no service finger
RouterA(config)#no ip identd
```

Services such as echo, discard, daytime, and chargen are considered TCP and UDP small services and can be disabled using **no service tcp-small-servers** and **no service udp-small-servers**. Some services might be needed, such as Simple Network Management Protocol (SNMP) and HTTP. These services will need to be tightened. The SNMP default community string must not be used and the new community string must be difficult to guess. The read-write access should be avoided at all costs and read-only access should be configured. Also, SNMP access should be restricted to certain known SNMP agents. Better yet, SNMP version 3 should be used instead. The topic of SNMP was discussed in Chapter 6, "Analyzing Network Data Traffic." The following is the example of the SNMP configuration on a Cisco router with read-only and with restricted access as defined in the access-list 10:

```
RouterA(config)#snmp-server community M@ke1tD1ff1cuLT ro 10
```

The Cisco IOS supports web-based remote administration, which is easier and more intuitive to use than the CLI mode that is used with telnet or SSH. However, HTTP has no encryption. Therefore, web-based administration via HTTP can reveal the passwords. This should be avoided just like Telnet. More recent Cisco IOS Software versions, starting from release 12.2(15)T, do provide another option of HTTPS that provides end-to-end SSL encryption, as shown:

```
RouterA(config)# no ip http server
RouterA(config)# ip http secure-server
RouterA(config)# ip http access-class 15
RouterA(config)# ip http authentication aaa
```

The configuration shows the normal HTTP service is disabled, and the HTTPS service is enabled instead. The HTTP access is also restricted with the access-list 15, and it will use the AAA authentication.

Besides disabling unnecessary services running on the router, some services or features that Cisco routers utilize should be disabled. It is highly recommended that services such as CDP, remote configuration downloading, and source-routing are disabled, as shown:

```
RouterA(config)#no cdp run
RouterA(config)#no service config
RouterA(config)#no ip source-route
```

Cisco equipment uses Cisco Discovery Protocol (**CDP**) to identify each other on a LAN segment. This feature is enabled automatically and it allows anyone on the network to collect the information of the network. The remote configuration is disabled by the command **no service config**. This stops the router from loading its configuration from the network, which is not secure. The routers are capable of loading their startup configuration from the local memory and this is more secure. Source-routing can be used in many kinds of attacks. By disabling this feature, the router will disregard the IP packet with source routes information.

CDP
Cisco Discovery Protocol. Proprietary protocol is used to obtain the platform and the protocol addresses of neighboring devices.

On the interface level, any unused interfaces should be disabled so that they cannot participate in any network activity. Directed broadcasts allow a host on another network segment to initiate a broadcast to a different network segment. This can be used as a denial of service attack like the smurf attack. This feature should be disabled. Newer IOS versions disable the directed broadcast by default.

The router interfaces should not be acting as the intermediary for ARP or ARP proxy. This feature will extend ARP traffic between the two network segments. This is not desirable and should be avoided. Also, the router can be used to relay ICMP messages that can be used by attackers, and the generation of these messages should be disabled on the interface. The common ICMP messages that are commonly exploited are Host unreachable, Redirect, and Mask reply. The following is the example configuration to secure the router interface:

```
RouterA(config)# interface fastethernet0/1
RouterA(config-if)#shutdown
RouterA(config)# interface fastethernet0/2
RouterA(config-if)# no ip directed-broadcast
RouterA(config-if)# no ip proxy-arp
RouterA(config-if)# no ip unreachables
RouterA(config-if)# no ip redirects
RouterA(config-if)# no ip mask-reply
```

Router Logging and Access-List

Logging is a critical part of security. Logging allows the administrator to analyze the events that occur and use the given information to correlate and find the issues. Cisco routers provide a great deal of logging events. They can log system errors, network and interface status, login access, access list matches, routing changes, and many more types of events. Cisco's log messages can be directed to the console,

terminal line, memory buffer, and syslog server. There are 8 levels (0–7) of log severity:

Emergencies (0)
Alerts (1)
Critical (2)
Errors (3)
Warnings (4)
Notifications (5)
Informational (6)
Debugging (7)

NTP
Network Time Protocol. A protocol that synchronizes the router's clock with the time server.

It is recommended that for best security, syslog logging and buffered logging in debugging level should be set up. To keep accurate logs, the correct time on a router has to be set up. This tends to be a step that many disregard. Cisco routers support the Network Time Protocol (**NTP**), which can be set up to synchronize the router's clock with the time server. To correlate the time with the log events, a timestamp service will need to be initiated. The following example shows the log configuration of a Cisco router. It consists of enabling timestamp service for log with the date and time down to millisecond detail. The logging can be enabled with the simple command **logging on**. A memory buffer of 16 Kbytes was reserved for logging with debugging level. In addition, the same debugging level log messages will be sent to a syslog server with the IP address 172.20.20.20:

```
RouterA(config)# service timestamps log datetime msec
RouterA(config)# logging on
RouterA(config)# logging buffered 16000 debugging
RouterA(config)# logging trap debugging
RouterA(config)# logging 172.20.20.20
```

The log information can be verified with the command **show log**, as demonstrated here:

```
RouterA#sh log
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering
disabled)
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 1 messages logged, xml disabled,
                    filtering disabled
    Logging Exception size (4096 bytes)
    Count and timestamp logging messages: disabled
No active filter modules.
    Trap logging: level debugging, 3882 message lines logged
        Logging to 172.20.20.20(global) (udp port 514, audit disabled,
link up), 5 message lines logged, xml disabled,
               filtering disabled

Log Buffer (16000 bytes):
```

```
*May 20 23:32:37.286: %SYS-5-CONFIG_I: Configured from console by
piyasat on vty1 (192.168.101.121)
```

Another way to use the log is via the access list. The router's access list and log can indeed be used in conjunction. As a matter of fact, it is a best practice to use log in every deny statement in each extended access list. This will provide valuable information of what is being denied, and it is useful as a security detection tool of probes and attacks against the network. The following is an access list statement with a keyword **log**:

```
access-list 101 permit ip 192.168.0.0 0.0.255.255 any
access-list 101 permit tcp any any eq 80
access-list 101 deny ip any any log
```

Typically, access lists vary from place to place depending on each entity's security policy. A type of traffic that is allowed on one network might not be desirable on another; however, there is a basic access list that is recommended for every network. This access list is used to protect against IP address spoofing. The IP address spoof protection concept is simple: Do not allow any inbound IP packets that contain IP source addresses of the internal network or any reserved private IP addresses. The following example shows an access-list 102 that contains an IP address spoofing protection for the internal network of 12.12.12.0/24. This access list will be applied to the outbound interface of the router as **ip access-group 102 in**:

```
access-list 102 deny ip 12.12.12.0 0.0.0.255 any log
access-list 102 deny ip 10.0.0.0 0.255.255.255 any log
access-list 102 deny ip 172.16.0.0 0.15.255.255 any log
access-list 102 deny ip 192.168.0.0 0.0.255.255 any log
access-list 102 permit ip any any
```

7-4 SWITCH SECURITY

Switches are another common network device. Typically, there are more switches than routers on a network. Switches are commonly used in the access layer of the network hierarchy. The access layer connects users who will share the common network resources and bandwidth. Directly interfacing with users can be a security challenge since there is no way to know what will be connecting to the switch access ports. The bottom line is that the better the network will be if more control or policies are enforced at the switch port level.

The previous section had covered the necessary steps to secure the routers. All discussed security steps or best practices, such as physical security, IOS updates, configuration hardening for local and remote access, disabling unnecessary services, logging and access lists apply to the switches as well. This is especially true on Cisco equipment because the IOS configuration is similar, if not the same, on the Cisco routers and switches. The major differences are with the interfaces commands. Some commands are reserved for only the router interface and some commands are reserved only for the switch interface. This section's focus is on manageable Cisco switches. Some of these concepts can apply to other switch vendors as well.

Switch Port Security

Since switch interfaces or ports directly connect users or network equipment, they should be securely configured to prevent malicious attacks or exploitations from end users. The very fundamental security rule is to disable the ports that are not being used. There are definitely more ports on switches than routers. A useful command to use when applying the same command to a group of switch ports is the range command. This will make it easier for the administrator to apply the same security policy on switch ports. The following example shows the commands to shut down a group or a range of interfaces:

```
SwitchA(config)# interface range Gigabitethernet1/1-24
SwitchA(config-if)#shutdown
```

switchport port-security
The command to enable the port security.

On Cisco switches, you can take advantage of a built-in set of switch port security commands. Switch port security can be configured to restrict a port's ingress traffic by limiting the MAC addresses that are allowed to send traffic into the port. The command to enable the port security is **switchport port-security**, and this command has to be issued at the interface level. A switch port can be configured to restrict access by setting a maximum number of MAC addresses or it can be configured to allow only known MAC addresses to pass traffic.

To configure the maximum number of MAC addresses on a switch port, use the command **switchport port-security maximum** [*number*]. To configure a port to allow certain MAC addresses to pass traffic, use the command **switchport port-security mac-address** [*mac_address*] or **switchport port-security mac-address sticky** [*mac_address*]. The big difference between the first **port-security mac-address** command and the second command with the **sticky** option is that the first command will not get the configured MAC address into the running configuration, while the second command with sticky option will. With the sticky option, the configured MAC address will appear in the running-config of the switch and it can then be saved into the startup-config. This way, when the switch reboots, the configured MAC address remains part of the configuration.

Violation Action
Defines the action taken if a switchport is violated.

protected
Drops packets from the violated MAC address(es).

restrict
The same as the protected mode, but it will also send SNMP trap messages to the SNMP server.

shutdown
This shutdowns the port and puts the port in ERRDISABLE state.

Along with the security configuration, one will need to define a **violation action**. When a violation occurs, one of the selected violation actions will take an effect. These violation actions are protected, restrict, and shutdown. The violation action **protected** will drop packets from the violated MAC address(es). The violation action **restrict** is the same as the protected mode, but it will also send SNMP trap messages to the SNMP server. The violation action **shutdown** is to shutdown the port and put the port in ERRDISABLE state. The following is an example of a port security configuration. The port security command **switchport port-security maximum 2** allows at most only two MAC addresses on this switch port Gigabitethernet 1/10. Not only that, only the specified MAC addresses of 0011.2233.440a and 0011.2233.440b are allowed to pass the traffic. If the violation occurs, the switch port will be shut down:

```
SwitchA(config)# interface Gigabitethernet1/10
SwitchA(config-if)#switchport port-security maximum 2
SwitchA(config-if)#switchport port-security mac-address sticky
0011.2233.440a
SwitchA(config-if)#switchport port-security mac-address sticky
0011.2233.440b
SwitchA(config-if)#switchport port-security violation shutdown
```

When a port is in the **ERRDISABLE** state, the port is automatically disabled by the switch operating system software because an error condition has been encountered on the port. This requires a manual intervention by the administrator in order to re-enable the port. The administrator will need to issue the command **shutdown** and then **no shutdown** to re-enable the port. Another way of re-enabling a port from ERRDISABLE state is to configure the errdisable recovery feature. The following example shows the errdisable discovery configuration to recover the port from port security violation after 10 minutes (600 seconds):

```
SwitchA(config)# errdisable recovery cause psecure-violation
SwitchA(config-if)#errdisable recovery interval 600
```

ERRDISABLE
In this state, the port is automatically disabled by the switch operating system software because an error condition has been encountered on the port.

Another port security feature that is not part of the **switchport port-security** command is **storm control**. The storm control feature can be used to limit the amount of unicast, multicast, or broadcast packets that each port can receive. When there is an excessive amount of any kind of these packets, it becomes a network storm. A network storm disrupts network services or degrades the network performance. The storm control feature is generally available on other switch vendors; however, their storm control features are limited only to broadcast and multicast. Cisco is one of the only few manufacturers to have unicast storm control. This is proven to be useful to help stopping denial of service attacks where a machine is transmitting an excessive amount of unicast packets.

storm-control
Used to limit the amount of unicast, multicast, or broadcast packets that each port can receive.

pps
Packets per second. A measure of the data transfer rate.

The **storm-control** command is applied at the switch port level. The following example shows the options that one could choose for unicast storm control. The storm can be suppressed by defining the bandwidth percentage of the interface or by defining the bps (bits per second) to limit the bandwidth the unicast traffic is allowed to consume. The storm can also be suppressed by defining the **pps** (packet per second) to limit the number of unicast packets a switch port can forward. Note that older Cisco switches might not be equipped with the pps option of the unicast storm:

```
SwitchA(config-if)#storm-control unicast level ?
  <0.00 - 100.00>  Enter rising threshold
  bps              Enter suppression level in bits per second
  pps              Enter suppression level in packets per second
```

The storm-control is configured with a **rising threshold** and a **failing threshold**. When the traffic rises above the rising threshold, the interface drops that specific traffic until the traffic comes down below the failing threshold. The administrator can enforce all types of storm controls as shown in the configuration of interface Gigabit Ethernet 1/0/8 here:

rising threshold/ failing threshold
When the traffic rises above the rising threshold, the interface drops that specific traffic until the traffic comes down below the failing threshold.

```
interface GigabitEthernet1/0/8
 storm-control broadcast level 2 1
 storm-control multicast level bps 50m 10m
 storm-control unicast level pps 6k 2k
 storm-control action trap
```

In this example configuration, the broadcast traffic is discarded if it uses more than 2 percent of the available bandwidth. This is 20 Mbps (2) for a Gigabit interface, and it will resume the broadcast traffic again if the traffic falls below 10 Mbps (1).

The multicast storm is set to use only 50 Mbps (50 m) and its threshold is set at 10 Mbps (10 m). The unicast storm is configured to discard any unicast packets above 6,000 (6 k) packets per second, and the unicast traffic will resume if it falls below the 2,000 (2k) packets per second. The default action for storm-control is to drop packets (trap) when the rising threshold is met. It can also be configured to shut-down an interface or to send SNMP trap messages.

Switch Special Features

The Spanning Tree Protocol (STP) is a common protocol found in every network switch. STP is a Layer 2 protocol designed to prevent loops within switched networks. STP builds its topology based on BPDU (Bridge Protocol Data Unit) messages. A vulnerability associated with STP is that a STP enabled device within the network can actively change the STP topology by sending an unexpected BPDU message. In order to prevent such events, features such as BPDU guard and BPDU filter can be used.

STP Portfast
This speeds up the STP process and transitions the port into a forwarding state bypassing the listen and learn states.

BPDU Guard
Feature used to prevent a STP Portfast to receive any BPDU message to modify the Spanning Tree topology.

On STP enabled switches, a switch port has to go through four STP states (i.e., block, listen, learn, and forward), before it can pass traffic. This process can take between 30 to 50 seconds. To reduce this time, a switch port can be configured to be an **STP Portfast**, which speeds up the STP process, and transitions the port into a forwarding state bypassing the listen and learn state. Typically, a STP Portfast interface is used to directly connect a host device, which does not send BPDU messages. **BPDU guard** is used to prevent a STP Portfast to receive any BPDU message to modify the Spanning Tree topology. Upon receipt of a BPDU, BPDU guard puts the interface configured for STP Portfast into the ERRDISABLE state. By default, BPDU guard is disabled. The following command is used to globally enable BPDU guard on all edge ports of a Cisco switch by default:

```
SwitchA(config)# spanning-tree portfast bpduguard default
```

BPDU guard can also be enabled at the interface level with the following command:

```
SwitchA(config)# interface gigabitethernet 0/1
SwitchA(config-if)# spanning-tree bpduguard enable
```

BPDU Filter
Feature that effectively disables STP on the selected ports by preventing them from sending or receiving any BPDU messages.

STP Root Guard
Feature that allows participation in spanning tree and BPDU messages as long as the attached device does not attempt to become the root bridge.

Another STP feature called **BPDU filter** offers a different flavor when dealing with BPDU. BPDU guard prevents a switch port from receiving any BPDU messages, but it does not prevent it from sending them. The BPDU filter feature effectively disables STP on the selected ports by preventing them from sending or receiving any BPDU messages. The switch port will ignore all BPDUs, and it will send no BPDUs. Similar to BPDU guard, the BPDU filter can be configured globally or on an individual port. The global command is **spanning-tree portfast bpdufilter default**. The command to enable the BPDU filter on the interface level is **spanning-tree bpdufilter enable**.

STP Root guard is another feature that can be used to protect the STP topology. Unlike the BPDU guard, STP Root guard allows participation in spanning tree and BPDU messages as long as the attached device does not attempt to become the root bridge. Essentially, STP Root guard provides a way to enforce the root bridge placement in the network. If an unauthorized device starts sending BPDU messages with a better bridge ID, the Root guard disables the switch port on which those

BPDU messages were received. The switch port will be in the ERRDISABLE state. The STP Root guard feature can only be enabled at the interface level. It is recommended to apply this feature to those switch ports that are not connected to the root bridge. The following is the command used within the interface configuration mode to enable STP Root guard:

```
SwitchA(config-if)# spanning-tree guard root
```

There are a few features that are specific to Cisco switches. It was recommended in the previous section that CDP should be disabled on the routers. This is the same on switches as well. Another Cisco proprietary feature is Virtual Trunking Protocol (VTP), which is a Layer 2 messaging protocol used to automatically add, delete, and rename VLANs on a network-wide basis. Using VTP allows for consistent VLAN configuration across all switches on the network. Cisco switches come with VTP enabled by default and they are enabled as VTP server mode. It is possible for a single switch to overwrite all VLAN assignments. It is recommended that VTP be disabled if it is not being used. The following is the example:

```
SwitchA(config)# no vtp mode
SwitchA(config)# no vtp password
SwitchA(config)# no vtp pruning
```

Another Cisco proprietary switch protocol is Dynamic Trunking Protocol (**DTP**), which is used to automatically negotiate a switch port to be either an access port or a trunk port. This is convenient, but at the same time, can be exploited to reveal all the VLANs or to gain access to all the VLANs. It is recommended that access ports and trunk ports should all be manually configured without negotiation. Also, it is best practice to control the number of VLANs that are allowed through the trunk port, as shown in the following example where VLANs 2, 3, and 30 are being allowed:

DTP
Dynamic Trunking Protocol. Used to automatically negotiate a switch port to be either an access port or a trunk port.

```
SwitchA(config)# interface gigabitethernet 0/1
SwitchA(config-if)# switchport mode trunk
SwitchA(config-if)# switchport trunk allowed vlan 2,3,30
SwitchA(config-if)# switchport nonnegotiate
```

7-5 WIRELESS SECURITY

This section provides an overview of securing 802.11 wireless LANs. The network administrator must be aware of the security issues when configuring a wireless LAN. The fact is, radio frequencies (RF) will pass through walls, ceilings, and floors of a building even with low signal power. Therefore, the assumption should never be made that the wireless data is confined to only the user's area. The network administrator must assume that the wireless data can be received by an unintended user. In other words, the use of an unsecured wireless LAN is opening a potential threat to network security.

To address this threat to WLAN security, the network administrator must ensure that the WLAN is protected by firewalls and intrusion detection, and most importantly the network administrator must make sure that the wireless security features are

TURNED ON! This might seem to be a bold statement, but surprisingly enough, many WLANs are placed on a network without turning on available wireless security features. Many times, the user in the WLAN assumes that no one would break into their computer because nothing important exists on the system. This may be true, but to an attacker, the user has one very important item—access to the wired network through an unsecured client.

WLANs use an **SSID** (service set identifier) to authenticate users, but the problem is that the SSID is broadcast in radio link beacons about 10 times per second. In WLAN equipment, the **beacons** are transmitted so that a wireless user can identify an access point to connect to. The SSID can be turned off so it isn't transmitted with a beacon, but it is still possible for the SSID to be obtained by packet sniffing. As noted previously, packet sniffing is a technique used to scan through unencrypted data packets to extract information. In this case, an attacker uses packet sniffing to extract the SSID from data packets. Disabling SSID broadcasting will make it so that most client devices (such as Windows devices and Mac devices) won't notice that the wireless LAN is present. This at least keeps "casual snoopers" off the network. Enterprise-grade access points implement multiple SSIDs, with each configured SSID having its own VLAN and wireless configuration. This allows the deployment of a common wireless LAN infrastructure that supports multiple levels of security, which is important for some venues such as airports and hospitals (where there are both public and private users).

IEEE 802.11 supports two ways to authenticate clients: open and sharekey. **Open authentication** basically is a null authentication that can enable any client to authenticate to an AP as long as the client knows the correct SSID. In **sharekey authentication**, both the client and the access point share a key called a pre-shared key (PSK). The client sends a shared key authentication request and then a packet of text called a challenge text is sent by the access point to the client with the instruction to encrypt the text and return it to the access point. This requires that wired equivalent privacy (**WEP**) be turned on. WEP is used to encrypt and decrypt wireless data packets. The exchange and the return of the encrypted text verify that the client has the proper WEP key and is authorized to be a member of the wireless network. Note that shared key authentication is extremely vulnerable. As a result, it's standard practice to avoid the use of shared key authentication. Figure 7-4 provides an example of the setting for WEP encryption. In part a of Figure 7-4, the user has the WEP options of disabled (No Privacy), 64-bit WEP (Privacy), and 128-bit WEP (More Privacy). Part b of Figure 7-4 shows the wireless security settings in Windows Vista. There are clearly more options, and these newer wireless security settings are discussed next.

It is well-known that WEP is a weak wireless security system. It doesn't use a strong enough encryption to secure a wireless network. The RC4 algorithm is used for encryption in WEP. A couple of the weaknesses of WEP include that the challenge text in WEP is sent in clear text. Additionally, the WEP initialization vector is only 24-bits in size and is always static. WEP also does not use a key management and its pre-shared key never changes. Because of these factors, it is not too difficult to obtain the pre-shared key. There is published information about WEP vulnerabilities, but even with this, WEP does provide some basic security and is certainly better than operating the network with no security.

SSID
Service set identifier. A 32 alphanumeric character unique identifier that's attached to data packets transmitted over a wireless network (WLAN). The SSID is essentially a password that enables the client to connect to the access point.

Beacon
Used to identify a wireless link.

Open Authentication
A null authentication that can enable any client to authenticate to an AP.

Sharekey Authentication
Both the client and the access point share a key called a pre-shared key (PSK).

WEP
Wired equivalent privacy. WEP provides a secure wireless channel by encrypting the data so that it is protected as it is transmitted from one end point to another.

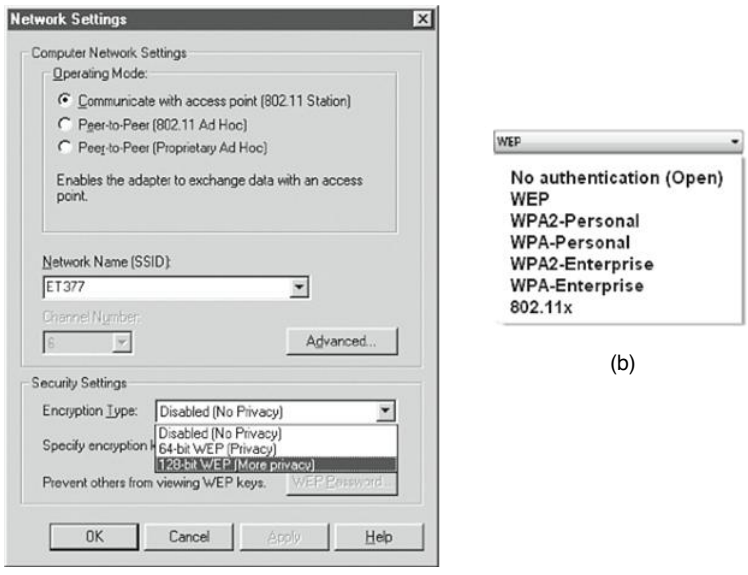

FIGURE 7-4 An example of setting WEP encryption on a wireless client

An improvement with wireless security is provided with WPA and WPA2. **WPA** stands for Wi-Fi Protected Access, and it supports the user authentication provided by 802.1x and replaces WEP as the primary way for securing wireless transfers. WPA still uses RC4 as the encryption algorithm, but it provides a key management mechanism via Temporal Key Integrity Protocol (**TKIP**). TKIP basically generates a sequence of WEP keys based on a master pre-shared key and rekeys periodically every 10,000 packets. TKIP also uses an integrity check value to ensure that the packet is not tampered with. If so, WPA will stop using the current key and will re-key. WPA2 is an improved version of WPA. It uses Advance Encryption Standard (**AES**) as its encryption algorithm and Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (**CCMP**) as its key management.

The encryption algorithm and key management alone cannot truly secure the wireless connection. The 802.1x standard enhances wireless security by incorporating authentication of the user. Cisco Systems uses an 802.1x authentication system called Lightweight Extensible Authentication Protocol (**LEAP**). In Cisco LEAP, the user must enter a password to access the network. This means that if the wireless client is being used by an unauthorized user, the password requirement will keep the unauthorized user out of the network.

WPA is considered to be a higher level of security for wireless systems. In the 802.1x system, a user requests access to the wireless network via an access point. The next step is for the user to be authenticated. At this point, the user can only send Extensible Authentication Protocol (**EAP**) messages. EAP is used in both WPA and WPA2 by the client computer and the access point. The access point sends an EAP message requesting the user's identity. The user (client computer) returns the identity information that is requested by the access point to an authentication

WPA
Wi-Fi Protected Access. Replaces WEP as the primary way for securing wireless transfers and it supports the user authentication provided by 802.1x.

TKIP
Temporal Key Integrity Protocol. Generates a sequence of WEP keys based on a master pre-shared key and rekeys periodically every 10,000 packets.

AES
Advance Encryption Standard. A 128-bit block data encryption technique.

CCMP
Counter Mode with Cipher Block Chaining Message Authentication Code Protocol. An encryption protocol designed for wireless LANs.

LEAP
Lightweight Extensible Authentication Protocol. A wireless security system used by Cisco.

EAP
Extensible Authentication Protocol. A general protocol used for supporting multiple authentication methods.

RADIUS
Remote Authentication Dial-In Service.

server. The server will then accept or reject the user's request to join the network. If the client is authorized, the access point will change the user's (client's) state to authorized. A Remote Authentication Dial-In User Service (**RADIUS**) service is sometimes used to provide authentication. This type of authentication helps prevent unauthorized users from connecting to the network. Additionally, this authentication helps to keep authorized users from connecting to rogue or unauthorized access points.

Another way to further protect data transmitted over a WLAN is to establish a VPN connection. In this way, the data is protected from an attacker. The following are basic guidelines for wireless security:

- Make sure the wireless security features are turned on.
- Use firewalls and intrusion detection on your WLAN.
- Improve authentication of the WLAN by incorporating 802.1x features.
- Consider using third-party end-to-end encryption software to protect the data that might be intercepted by an unauthorized user.
- Whenever possible, use encrypted services such as SSH and Secure FTP.

The bottom line is that the choice of the level of security will be based on multiple factors within the network. For example, what is the cost benefit ratio of increased security? How will incorporating or not incorporating increased wireless security affect users? The network administrator and the overall management will have to make the final decision regarding wireless security before it is installed and the network becomes operational.

7-6 VPN SECURITY

VPN
Virtual private network. Enables the remote clients to become part of the trusted network by establishing a secure connection between the remote end and the private network.

When a network is protected behind the firewall, it is sometimes referred to as a *private* network. Only computers on the same private network are considered to be trusted. Public access to this kind of network can be very limited. Access to a private network requires special permission to be granted on the firewall. Imagine a sales company that has its sales workforce throughout the country. The salespeople need to access the company's servers and databases at its headquarters, which is protected behind a firewall. It would be a network administrator's nightmare to grant individual access through the company's firewall. This idea does not allow for flexibility and mobility. Virtual private network (**VPN**) offers a solution to this problem. As the name implies, VPN is a concept of extending a private or a trusted network over public infrastructure like the Internet. A VPN accomplishes this by establishing a secure connection between the remote end and the private network, therefore enabling the remote clients to become part of the trusted network.

A secure VPN connection between two endpoints is known as an **IP tunnel**. A tunnel is created by an encapsulation technique, which encapsulates the data inside a known protocol (IP) that is agreed upon by the two end points. A tunnel creates a virtual circuit-like between the two endpoints and makes the connection appear like a dedicated connection even though it spans over the Internet infrastructure. Two types of VPNs are commonly used today:

- **Site-to-site VPN**: Used to create a virtual link from one site to the other. It essentially replaces the traditional WAN-type connection used in connecting typical sites. This type of VPN requires network hardware like a router or a firewall to create and maintain the connection.
- **Remote-access VPN**: Used to facilitate network access for users in remote office networks or for remote users that travel a lot and need access to the network. The client usually initiates this type of VPN connection.

IP Tunnel
An IP packet encapsulated in another IP packet.

VPN Tunneling Protocols

This section provides a quick overview of the protocols used in the creation of these VPN tunnels. One of the original tunneling protocols is the Generic Routing Encapsulation (**GRE**). GRE was developed by Cisco in 1994 and is still being used today. GRE is commonly used as a site-to-site VPN solution because of its simplicity and versatility. It is the only tunneling protocol that can encapsulate up to 20 types of protocols. In the past, when protocols like AppleTalk, Novell IPX, and NetBEUI roamed the network, GRE was the tunneling protocol of choice to carry these protocols to other remote sites.

GRE
Generic Routing Encapsulation. A simple IP packet encapsulation protocol developed by Cisco System, commonly used as a site-to-site VPN solution because of its simplicity and versatility.

Establishing a GRE tunnel through the IP telco cloud to connect Router A with Router C requires that the source and destination addresses of the physical network connection be defined as shown in Figure 7-5; Router A connects to the telco cloud via the router's Serial 0/1 interface. The IP address of 192.168.210.5 with a subnet mask of 255.255.255.0 has been assigned to the Router A Serial1 interface. Router C (the remote router) connects to the telco cloud via its Serial 0/1 interface. The IP address assigned to Router C's Serial1 interface is 192.168.100.3 with a subnet mask of 255.255.255.0. (*Note:* Any interface that connects to the telco cloud can be used to set up the VPN interface.) A tunnel is next established on each of the routers. The tunnel is assigned an IP address that is used in the home network. For example, the home network is a 10.0.0.0 network; therefore, the tunnel between Router A and Router C will have a 10.x.x.x IP address. The tunnel between Router A and Router C is called *tunnel 0* and is assigned IP addresses of 10.10.30.1 and 10.10.30.2. After the tunnel has been created across the IP network, the two routers appear to be on the same 10.10.30.x network.

The tunnel connection makes remote users appear as if they are part of the home network. This is accomplished by encapsulating the IP packet. The first packet uses the IP address that the remote user will use after the connection has been made. The encapsulation is used to transport the data across the networks.

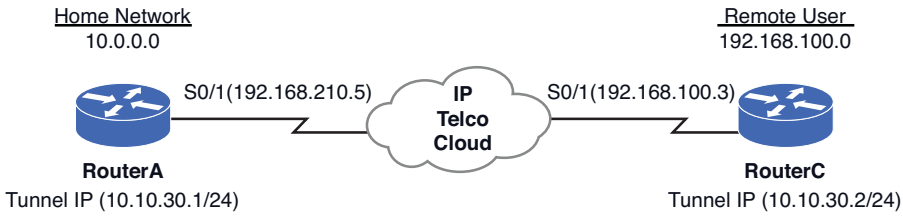

FIGURE 7-5 An example of a GRE tunnel through the IP telco cloud

Figure 7-6 provides an example of tunnel encapsulation. Part a of Figure 7-6 shows the basic IP packet. The source and destination IP address for the packet are listed. The source IP address for the remote tunnel is 10.10.30.2. The IP tunnel address for the home router interface is 10.10.30.1. This is the destination IP address. Part b of Figure 7-6 shows the layer of encapsulation for the VPN tunnel. The source IP address of 192.168.100.3 is for Router C's Serial 0/1 interface. The destination IP address of 192.168.210.5 is for the home network's Router A Serial 0/1 interface.

The 192.168.100.3 and 192.168.210.5 IP addresses are used to deliver the VPN encapsulated data packets over a TCP/IP network. When a packet is delivered to the destination, the encapsulation layer is removed and the data packet will appear as if it came from the home 10.10.30.x network. Remember, the original packet, shown in part a of Figure 7-6, contains the IP addresses for the VPN tunnel, and the encapsulation layer [part b of Figure 7-6] includes the actual physical interface IP addresses for the VPN tunnel.

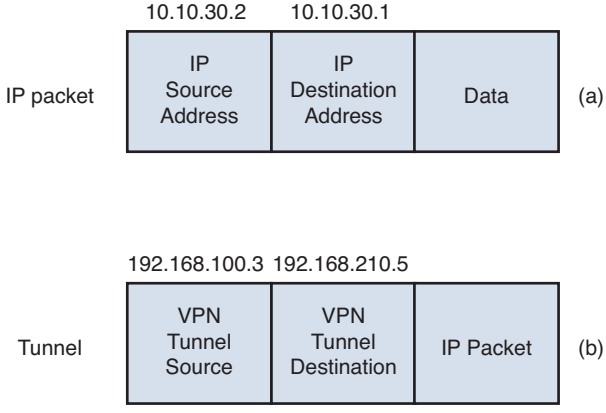

FIGURE 7-6 Data encapsulation for VPN data packets

Configuring a VPN Virtual Interface (Router to Router)

In this example, a VPN virtual interface will be configured on two routers on two separate networks. Figure 7-7 provides an illustration of the VPN. This example simulates a situation in which a VPN virtual interface is to be established between

two networks. This requires that a virtual interface be established on the home router, Router A, and the remote user on Router C in Figure 7-7. The tunnels on each router will be called *tunnel 0*. These interfaces each have their own IP addresses. The difference is the tunnels require a source and destination IP address for the tunnel. The source and destination addresses are the physical addresses (IP addresses) for the serial interfaces on each router that connect to the telco cloud.

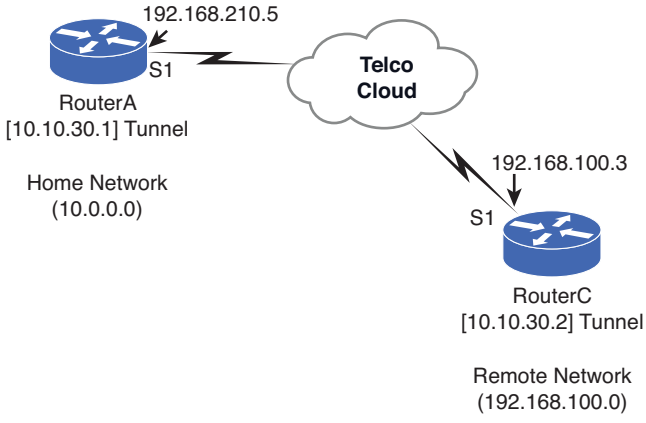

FIGURE 7-7 A virtual private network

Make sure routing has been properly configured for each router in the network, the interfaces have been assigned an IP address, and the destination physical interface from each end can be pinged prior to setting up the VPN interface and the tunnel. For example, Router A can ping the remote network router (Router C) using the command **ping 192.168.100.3**. Remember, the IP address for the serial interface is being pinged, and the virtual interfaces have not yet been configured:

```
RouterA#ping 192.168.100.3
Type escape sequence to abort.
Sending 5 100-byte ICMP Echos to 192.168.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max 52/56/72 ms
```

The successful ping indicates that there is a network connection between Router A on the home network (192.168.210.0) and Router C on the remote network (192.168.100.0). The next step is to configure the tunnel to establish a virtual private network between the two networks.

A tunnel is added to the router from the configuration mode **(config)#**. The command **int tunnel0** is entered to configure the tunnel interface. This places the router in the interface configuration mode—**(config-if)#**. The next step is to configure the virtual IP address to be used by the VPN tunnel. The IP address for the VPN tunnel must be on the same subnet as the home network IP address. For example, the home

network is from the 10.0.0.0 network. In this case, the tunnel 0 interface on Router A will be assigned an IP address of 10.10.30.1. The following shows the two steps for configuring the IP tunnel:

```
RouterA(config)#int tunnel0
RouterA(config-if)#ip 10.10.30.1 255.255.255.0
```

The next step is to define the IP destination and source address for the tunnel created on the home network router, Router A. Referring to Figure 7-7, the source IP address for tunnel 0 from Router A is 192.168.210.5 and the destination IP address for the remote interface on Router C is 192.168.100.3. After configuring the tunnel source and destination IP addresses, the router prompts that the "line protocol on Interface Tunnel0" changed state to **up**:

```
RouterA(config-if)#tunnel destination 192.168.100.3
RouterA(config-if)#tunnel source 192.168.210.5
00:31:37: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0,
changed state to up
```

You can use the command **sh ip int brief** to check the configuration for the tunnel interface as shown. A tunnel will show status **up** and protocol **up** even if it is not actually up. In this case, the other end of the tunnel connection has not yet been configured. The only way to really test the virtual link is with a **ping**. This will be demonstrated after the remote router (Router C) is configured:

```
RouterA#sh ip int brief
Interface        IP-Address      OK?  Method  Status   Protocol
FastEthernet0/0  10.10.20.250   YES  NVRAM   up       up
Serial0/0        10.10.100.10   YES  NVRAM   up       down
Serial0/1        192.168.210.5  YES  NVRAM   up       up
Tunnel0          10.10.30.1     YES  manual  up       up
```

The Serial1 interface shows status **up** and protocol **up**. This is the connection to the communications carrier (telco), and this interface must be up for the tunnel to work. The command **sh int tunnel 0** can be used to check to see whether tunneling has been configured on the router:

```
RouterA#sh int tunnel 0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.10.30.1/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
Encapsulation TUNNEL, loopback not set, keepalive set (10 sec) Tunnel
source 192.168.210.5, destination 192.168.100.3
Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
Checksumming of packets disabled, fast tunneling enabled
Last input never, output 00:00:01, output hang never Last clearing of
"show interface" counters never Queueing strategy: fifo
Output queue 0/0, 0 drops; input queue 0/75, 0 drops
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
```

```
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
20 packets output, 1200 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
```

The following output is a portion of the configuration file displayed using the **show run** command. This output shows that the tunnel has been configured on Router A:

```
RouterA#sh run
.
.
!
!
interface Tunnel0
ip address 10.10.30.1 255.255.255.0
 no ip directed-broadcast tunnel source 192.168.210.5
 tunnel destination 192.168.100.3
!
!
RouterC#conf t
.
```

The next step is to configure the remote user on Router C. First, enter the router's configuration mode using the **conf t** command. Remember, Router C's connection to the telco cloud has already been established and a connection verified using the **ping** command. Recall that the IP address of 192.168.100.3 was pinged from Router A. Next, enter the configuration for the tunnel 0 interface using the **int tunnel 0** command. At the **(config-if)#** prompt, enter the source and destination addresses for tunnel 0 as shown in the following code:

```
Enter configuration commands, one per line. End with CNTL/Z.
RouterC(config)#int tunnel 0
RouterC(config-if)#ip address 10.10.30.2 255.255.255.0
RouterC(config-if)#tunnel destination 192.168.210.5
RouterC(config-if)#tunnel source 192.168.100.3
```

The configuration for the interfaces can be checked using the **sh ip int brief** command as shown in the following code. Note that the tunnel is listed as a separate interface on the router:

```
RouterC#sh ip int brief
Interface       IP-Address    OK? Method  Status                Protocol
FastEthernet0/0 unassigned    YES not set administratively down down
Serial0/0       192.168.100.3 YES manual  up                    up
Serial0/1       unassigned    YES not set administratively down down
Tunnel0         10.10.30.2    YES manual  up                    up
```

The following output is a portion of the configuration file displayed using the **show run** command. This output shows that the tunnel has been configured on Router C:

```
RouterC#sh run
.
.
!
!
interface Tunnel0
ip address 10.10.30.2 255.255.255.0
 no ip directed-broadcast tunnel source 192.168.100.3
 tunnel destination 192.168.210.5
!
!
.
.
```

Each end of the VPN has now been configured. The next step is to check the link using the **ping** command. The VPN tunnel virtual address of 10.10.30.2 is pinged from computer A, as shown next:

```
RouterA#ping 10.10.30.2
Type escape sequence to abort.
Sending 5 100-byte ICMP Echos to 10.10.30.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max 80/80/84 ms
```

The **ping** shows that a virtual connection has been established between the remote user and the home network router. An interesting test of the network is to compare the results of running a **traceroute** from the home network to the remote user using the IP address for the remote user's router interface [192.168.100.3] and then running a traceroute from the home network to the remote user's VPN tunnel address [10.10.30.2]. The results of the test are shown here:

Traceroute to the physical interface IP address:

```
RouterA#trace 192.168.100.3
Type escape sequence to abort. Tracing the route to 192.168.100.3
1 192.168.210.5 16 msec 16 msec 16 msec
2 192.168.100.3 32 msec 32 msec *
```

Traceroute to the VPN tunnel IP address:

```
RouterA#trace 10.10.30.2
Type escape sequence to abort.
Tracing the route to 10.10.30.2
1 10.10.30.2 48 msec 48 msec *
```

The results of the first trace show that it takes two hops to reach the destination. In fact, a **traceroute** to the physical IP address will typically show multiple router hops. The test on the VPN tunnel shows that the trace took only one hop because the remote end of the VPN tunnel is configured to be directly connected to the

home 10 network. The **traceroute** on a VPN tunnel from a home router to the remote user will show only one hop.

This section has described how to configure a VPN tunnel link. The commands for configuring and verifying the link have been discussed. The following describes steps for troubleshooting a VPN tunnel link if problems should occur.

Troubleshooting the VPN Tunnel Link

The following steps are used to verify that the VPN tunnel is properly configured:

1. Confirm connection to the physical interface's IP address using a **ping**.
2. Check the source and destination IP addresses of the tunnel configured on the router.
3. Make sure the IP addresses on the ends of the tunnel are in the same subnet.
4. **Ping** the destination from the source.
5. Use **show run** to make sure the source and destinations are properly configured.

This example has shown how a VPN tunnel can be established using two routers. This situation is appropriate for establishing a permanent VPN connection to a remote user, such as a remote office for a company. However, a remote user who travels will have to establish a VPN connection directly from his or her PC, through an ISP and the VPN server in the home network. The tunneling protocols commonly used in remote access VPNs are mentioned throughout the rest of this section. To better understand remote-access VPN, you should at least understand the importance of Point to Point Protocol (**PPP**). In the days when modems and dial-ups were kings, PPP was the key to the remote access solution; it was the de facto protocol of the dial-up networking. In those days, people would make a dial-up connection to their ISP and establish a PPP session to the Internet. Even though authentication is optional for PPP, most implementations of PPP provide user authentication using protocols like Password Authentication Protocol (**PAP**) or Challenge Handshake Authentication Protocol (**CHAP**). PAP is a simple, clear-text (unencrypted) authentication method, which is superseded by CHAP, an encrypted authentication method that uses the **MD5** hashing algorithm. Later, Extensible Authentication Protocol (**EAP**) was introduced as another PPP authentication method. During the PPP authentication phase, the ISP dial-up server collects the user authentication data and validates it against an authentication server like a **RADIUS** server. RADIUS stands for Remote Authentication Dial-In User Service. RADIUS is an IETF standard protocol that is widely used for authenticating remote users and authorizing user access. The RADIUS server supports many methods of user authentication including PAP, CHAP, and EAP. Even though PPP dial-up is not as prevalent today, the concepts of central authentication still lend themselves to many technologies and applications.

Point-to-Point Tunneling Protocol (**PPTP**) was developed jointly by Microsoft, 3Com, and Alcatel-Lucent in 1996. It has never been ratified as a standard. Microsoft was a big advocate of PPTP and made PPTP available as part of Microsoft Windows Dial-up Networking. A PPTP server was included in Microsoft NT 4.0 server, and PPTP was widely used as a remote access solution. PPTP was designed

PPP
Point to Point Protocol. The de facto protocol of the dial-up networking.

PAP
Password Authentication Protocol. A simple, clear-text (unencrypted) authentication method.

CHAP
Challenge Handshake Authentication Protocol. An encrypted authentication method that uses the **MD5** hashing algorithm.

MD5
Message Digest 5. A cryptographic hash function that produces a 128-bit (16-byte) hash value.

EAP
Extensible Authentication Protocol. Introduced as another PPP authentication method.

RADIUS
Remote Authentication Dial-In User Service. Widely used for authenticating remote users and authorizing user access.

PPTP
Point to Point Tunneling Protocol. Designed to work in conjunction with a standard PPP.

to work in conjunction with a standard PPP. PPTP client software would establish a PPP connection to an ISP, and once the connection is established, it would then make the PPTP tunnel over the Internet to the PPTP server. The PPTP tunnel uses a modified GRE tunnel to carry its encapsulated packet for IP transmission. The diagram of typical PPTP connection and other tunneling protocols is represented in Figure 7-8. PPTP does not have any authentication mechanism, so it relies heavily on the underlying PPP authentication.

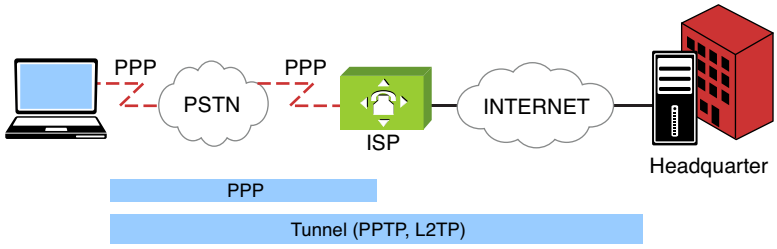

FIGURE 7-8 Tunneling diagram of PPTP and L2TP

L2F
Layer 2 Forwarding.

Layer 2 Forwarding Protocol (**L2F**) was developed by Cisco around the same time as PPTP. L2F was not used widely in the consumer market due to its requirement of L2F hardware. Unlike PPTP, where the VPN client software is installed and initiated from the client, L2F does not require any VPN client software. An L2F connection is intended to be done by L2F hardware. This hardware is designed to be at the ISP. A client would make a typical PPP connection to the ISP. The ISP will then initiate the L2F tunnel connection on UDP port 1701 to the L2F server at the corporate headquarters. This requires coordination between the ISP and the corporate network. L2F relies on the PPP authentication to be passed on to the corporate authentication server.

L2TP
Layer 2 Tunneling Protocol.

AH
Authentication Header. A security protocol used by IPsec that guarantees the authenticity of the IP packets.

ESP
Encapsulating Security Protocol. A security protocol used by IPsec that provides confidentiality to the data messages (payloads) by way of encryption.

Layer 2 Tunneling Protocol (**L2TP**) was developed by the Internet Engineering Task Force (IETF) in 1999. L2TP was created with the intention of merging two incompatibles proprietary tunneling protocols, PPTP and L2F. L2TP is considered to be an enhancement of the two previous protocols. L2TP does not require a specific hardware. It can be initiated directly from the client. L2TP Tunnel encapsulation is done on UDP port 1701. L2TP allows for tunnel authentication, so it does not have to rely heavily on the underlying PPP. If L2TP is used over an IP network where PPP is not used, the tunnel can be created with its own authentication mechanism.

All of the previously mentioned tunneling protocols are lacking one important security feature—encryption. Encryption can guarantee data confidentiality in the tunnel. IPsec offers encryption features that the others lack. IPsec was designed for the purpose of providing a secure end-to-end connection. The VPN can take advantage of IPsec to provide network layer encryption and authentication techniques. IPsec is versatile in that it can be implemented easily as a remote access VPN or as a site-to-site VPN. For IPv6, IPsec becomes an even more integral part as it is embedded within the IPv6 packets. There are two primary security protocols used by IPsec: Authentication Header (**AH**) and Encapsulating Security Payload (**ESP**). AH guarantees the authenticity of the IP packets. It uses a one-way hash algorithm, like

Message Digest 5 (MD5) or Secure Hash Algorithm 1 (**SHA-1**), to ensure the data integrity of the IP packets. ESP provides confidentiality to the data messages (payloads) by way of encryption. It uses symmetrical encryption algorithms like Data Encryption Standard (**DES**), Triple Data Encryption Standard (**3DES**), and Advanced Encryption Standard (**AES**).

Before an IPsec tunnel can be established, quite a few security parameters have to be negotiated and agreed upon by both ends. IPsec uses the Internet Key Exchange (**IKE**) protocol to manage such a process. IKE is a hybrid protocol that encompasses several key management protocols, most notably Internet Security Association and Key Management Protocol (**ISAKMP**). Many times, the term IKE and ISAKMP are mentioned alongside each other. There are two negotiation phases that the two network nodes must perform before the IPsec tunnel is complete. The IKE Phase 1 is a phase where both network nodes authenticate each other and set up an IKE SA (Security Association). In phase 1, the **Diffee-Hellman** key exchange algorithm is used to generate a shared session secret key to encrypt the key exchange communications. This phase is essentially to set up a secure channel to protect further negotiations in phase 2. The following is an example configuration of the IKE Phase 1 or the IKE policy on Cisco routers.

```
crypto isakmp policy 1
 encr 3des
 hash md5
 authentication pre-share
crypto isakmp key PJtcF7yF2w address 172.16.117.14
```

IKE Phase 2 uses the secure channel established in phase 1 to negotiate the unidirectional IPsec SAs—inbound and outbound—to set up the IPsec tunnel. This is where the algorithm parameters for AH and ESP would be negotiated. The following is an example configuration of the IKE Phase 2 or the Crypto map policy on Cisco routers. Both the AH and ESP will be using SHA with HMAC variant as the authentication algorithm protocol. To encrypt the payload, the ESP uses 3DES as the encryption algorithm. These parameters are entered as the transform-set called "AH-and-ESP" and will be applied to the IPsec to tunnel. The IPsec tunnel configuration itself consists of the IP address of its peer or the other end of the tunnel, the transform-set and the access-list allowing only certain networks to transverse through the secured tunnel:

```
crypto ipsec transform-set AH-and-ESP ah-sha-hmac esp-3des esp-sha-
hmac
!
crypto map IPSec_Phase2 1 ipsec-isakmp
 description Tunnel to 172.16.117.14
 set peer 172.16.117.14
 set transform-set AH-and-ESP
 match address 104
!
```

SHA-1
Secure Hash Algorithm. It is used to ensure the data integrity of the IP packets.

DES, 3DES
Data Encryption Standard, Triple Data Encryption Standard. A symmetrical encryption algorithm.

AES
Advanced Encryption Standard. A symmetrical encryption algorithm.

IKE
Internet Key Exchange. A hybrid protocol that encompasses several key management protocols, most notably Internet Security Association and Key Management Protocol.

ISAKMP
Internet Security Association and Key Management Protocol. A protocol for establishing Security Associations (SA) and cryptographic keys in an Internet environment.

Diffee-Hellman
Key generation algorithm. Used to generate a shared session secret key to encrypt the key exchange communications.

SUMMARY

This chapter examined the topics of network security. The concept of denial of service (DoS) was examined in Section 7-1. This section also examined the SYN attack, smurf attack, and Distributed Denial of Service (DDoS) attack. Section 7-2 examined firewalls and access lists. Topics included in the section included stateful firewalls, demilitarized zones, and configuring access lists. Sections 7-3 through 7-6 provided a look at "best practices" for setting up security on routers, switches, and wireless networks. This chapter concluded with a look at configuring external access to networks using a VPN.

QUESTIONS AND PROBLEMS

Section 7-1

1. What is a denial of service attack?
2. Describe a SYN attack.
3. Cisco routers use what command to block broadcasts to a subnet?
4. Define a directed broadcast.
5. What is the best way to keep from contributing to DDoS attacks?
6. What is a smurf attack?

Section 7-2

7. What is the purpose of a firewall?
8. Why is a stateful firewall important?
9. What is the router command for setting an access list 100 to block SNMP UDP packets from any source to any destination?
10. What command should be added to the end of an access list 150 to allow other data packets to enter and exit the router?
11. An extended IP access list 130 is to be applied to a router's Serial0 interface. The list is to be applied to inbound data packets. What command should be entered from the router's (config-if)# prompt?
12. Modify the command for problem 11 so that the access list is applied to outbound data packets.
13. What command is used to view the number of times an access list has been matched?
14. It has been determined that a computer with an IP address of 192.168.8.4 is flooding the network with ICMP packets. Create an extended access list to stop the flood.

15. Apply the access list created in problem 14 to the inbound data traffic on the router's Serial1 interface.
16. What is the purpose of the demilitarized zone and why is this used?
17. What are first two steps for applying access lists on a router?
18. What is the purpose of the following command in JUNOS?

```
net-admin@noc# set term block-snmp from destination-port 161
```

19. What is the purpose of the following command in JUNOS?

```
net-admin@noc# set term allow-snmp from protocol udp
```

20. What is a prefix list in JUNOS?
21. What command is used in JUNOS to verify the filter list? List the prompt and the command.

Section 7-3

22. What is always on top of the list of any "best security" practice, and why is this important?
23. Why is it important to keep the operating system of a router up to date?
24. What are the two types of passwords on a router?
25. What does the following command do?

```
RouterA(config)# username admin privilege 10 password @dm1np@$$wd
```

What is level 10?

26. What is the purpose of configuring AAA on a router?
27. What does the following command do?

```
RouterA(config)#crypto key generate rsa
```

28. What does the command **transport input ssh** do?
29. What commands can be used to disable services like echo, discard, daytime, and chargen? List the prompt and the command used for this.
30. What does the following command do?

```
RouterA(config)#snmp-server community M@ke1tD1ff1cuLT ro 15
```

31. What is the purpose of the following commands, and why is it important to use these?

```
RouterA(config)#no cdp run
RouterA(config)#no service config
RouterA(config)#no ip source-route
```

32. What command is used to enable logging on a router?
33. What is the purpose of the network time protocol?
34. Prepare an access-list 102 that can be used to prevent spoofing.

Section 7-4

35. What single command can be used to shut down multiple switch ports? Assume that there are 24 ports. List the command and the prompt.

36. What command is used to enable port security on a switch? List the command and the prompt.

37. What does the following command do?

```
SwitchA(config-if)#switchport port-security maximum 2
```

38. What command can be used on a switch to limit the amount of unicast packets that each port can receive if the rising threshold is 15 MBps and the Falling threshold is 5 MBps.

39. What does the following command do?

```
storm-control unicast level pps 7k 3k
```

40. What are the four states an STP enabled switch goes through before it can pass data traffic?

41. What does the command **spanning-tree bpdufilter enable** do on a switch?

42. What command is used to enable STP Root guard. List the command and the prompt.

43. What does the following command do?

```
SwitchA(config-if)# switchport trunk allowed vlan 5,7,18,20
```

Section 7-5

44. What is the most important thing to do if using a wireless network?

45. What is the purpose of wireless beacons?

46. What information can be obtained from a wireless beacon?

47. What is the purpose of WEP?

48. List five guidelines for wireless security.

49. Describe the steps used by WPA2 to authenticate a user.

50. What is a RADIUS server?

Section 7-6

51. What is the goal of a VPN tunnel?

52. Draw a sketch of the encapsulation of a VPN data packet. Show the IP source and destination address and the VPN tunnel source and destination address encapsulated with the IP packet.

53. Explain the expected difference when running a traceroute from the home network to the remote user using the IP address for the remote user's router interface, and then running a traceroute from the home network to the remote user's VPN tunnel address.

54. List five steps for troubleshooting the VPN tunnel link.

55. Identify two tunneling protocols that can be used to configure a remote user's PC.

56. What does encryption guarantee?

57. What are the two primary security protocols used by IPsec?

58. What is IKE?

59. List the command and the router prompt for configuring an IP tunnel 0 to 172.16.25.1 using a subnet mask of 255.2555.255.0.

60. What does the following command do?

```
RouterA(config-if)#tunnel destination 192.168.200.5
```

Critical Thinking

61. Your network is experiencing an excessive amount of pings to your network server. The pings are from outside the network. Someone suggests that you set an access list to block ICMP packets coming into the network. How would you respond?

62. Your supervisor informs you that a user on the network has requested a VPN connection. Prepare a response to the supervisor discussing what is needed to provide the connection.

63. Provide an example of configuring AAA on a Cisco router. Can authentication be applied locally, remotely, or both?

64. Your task is to configure VTY remote access with SSH. The configuration uses the default login authentication that was defined by AAA. Issue the command to enforce that SSH is the only access method. Use access-class of 12 that defines the access-list of the network that is allowed to connect to the router via SSH. Lastly, the exec-timeout of 10 minutes is configured. List the commands and corresponding prompt to accomplish this.

65. A memory buffer of 16 Kbytes is reserved for logging with debugging level. In addition, the same debugging level log messages will be sent to a syslog server with the IP address of 192168.10.10. List the commands used to enable this feature.