This chapter covers the following topics:

- Physical Security Issues
- Layer 2 Security Considerations
- IP Addressing Design Considerations
- ICMP Design Considerations
- Routing Considerations
- Transport Protocol Design Considerations
- DoS Design Considerations

CHAPTER 6

General Design Considerations

Many things difficult to design prove easy to performance. —Samuel Johnson, *Rasselas: The History of Rasselas, Prince of Abissinia*, 1759

A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible. There are no prima donnas in engineering. —Freeman Dyson, Physicist, *Disturbing the Universe*, 1979

At the beginning of any secure network design project, many best practices apply more or less uniformly to all areas of the design. This chapter presents these practices in a single location and then draws on them throughout the rest of the book. The designs presented in Chapter 13, "Edge Security Design," Chapter 14, "Campus Security Design," and Chapter 15, "Teleworker Security Design," are based on many of the concepts described here and in the companion chapters (Chapters 7–11), which detail specific design considerations for certain technologies. The topics are presented in loose compliance with the seven-layer OSI model and, as such, cover a diverse set of topics. Chapter 1, "Network Security Axioms," presented the security axioms; this chapter translates them into actionable guidance for secure network design.

Physical Security Issues

One common security truism is "Once you have physical access to a box, all bets are off." This is a good beginning assumption for this section. If an attacker has physical access to a computer, router, switch, firewall, or other device, your security options are amazingly limited. Networking devices, with few exceptions, can have their passwords reset by attaching to their console port. Hosts can be booted with a special floppy disk or CD-ROM designed to circumvent most host security on the device.

This book does not cover physical security issues in detail. Topics such as disaster recovery, site selection, and so on are not discussed at all. However, as a network designer, you must know where you are relying on physical security to augment or support your network security. There are some rules you can follow to improve your security:

- Control physical access to facilities.
- Control physical access to data centers.
- Separate identity mechanisms for insecure locations.

- Prevent password-recovery mechanisms in insecure locations.
- Be aware of cable plant issues.
- Be aware of electromagnetic radiation.
- Be aware of physical PC security threats.

The rest of this section examines these seven areas.

Control Physical Access to Facilities

Effectively controlling physical access to your organization's facilities should be the single top concern for both your physical security staff and you, the network designer. Most organizations utilize one of three mechanisms to implement physical security (presented in increasing order of security):

- Lock-and-key access
- Key card access
- Key card access with turnstile

Lock-and-Key Access

The most common physical security control, particularly in smaller organizations, is traditional lock-and-key access. For this method, individuals who need access to certain rooms or buildings are given keys for access. This option has the following benefits:

- Generally, this is the cheapest option for small organizations.
- No technical experience is required.
- Special keys are available to thwart key duplication.

However, there are also several drawbacks:

- If employees leave the company on less than amicable terms, they might "lose" their keys or might simply stop showing up for work. In such cases, it can be very costly to rekey the locks and redistribute keys to the valid employees.
- Unless coupled with an alarm system that augments the lock-and-key access, there is no mechanism to determine when employees with keys access a given physical location.
- Most keys can be easily duplicated at the local hardware store.
- Key authentication is *single-factor,* meaning the key is all a person needs to access locked areas.

Key Card Access

More common in larger organizations, key card access can alleviate some of the management problems associated with lock-and-key access and can provide increased security measures. Key card access can take the form of a magnetic card reader or a smart card. All of these systems have the same basic pros and cons once you eliminate the technical differences of the technology. These are the benefits of a key card system:

- Access to multiple locations can be controlled with a single card.
- In the event that an employee leaves the company, the employee's card can be quickly disabled whether or not it is physically returned.
- Locks should never need to be "rekeyed."
- Facilities with multiple entrances are easily supported.
- Reports can be run to show when individuals entered specific locations.

The drawbacks to a key card system are as follows:

- Like lock-and-key access, key cards are single-factor security. Any individual with a valid key card could access the location.
- Key card systems can be expensive, and in the event of a failure in the central authentication system, all users can be denied access to a facility.
- The principal problem with key card access is tailgating. *Tailgating* is gaining unauthorized access to a building by following an individual with valid access. Oftentimes, if attackers are dressed in the appropriate clothing, they can simply follow legitimate individuals into a building without having to present a key card. Even if someone requests to see a card, an attacker can show an invalid card because it might not actually be scanned by the card reader.

Key Card Access with Turnstile

Although most often associated with ballparks and stadiums, turnstile access with a key card can be one of the most secure methods of controlling physical access to a building. For this method, a key card is used to activate the turnstile and allow one person into the building. These systems are most common in large multifloor buildings, where access can be controlled at the ground floor. In the following list, you can see that this option has all the benefits of the previous option plus more.

- Tailgating is greatly diminished because only one person can enter per card.
- Access to multiple locations can be controlled with a single card.
- In the event that an employee leaves the company, the employee's card can be quickly disabled whether or not it is physically returned.
- Locks should never need to be "rekeyed."

- Reports can be run to show when individuals enter specific locations.

The drawbacks of a system such as this are as follows:

- Like the previous two systems, key card access with turnstile is a single-factor identity system. Any individual with a valid card could gain access to the building.
- This doesn't work well for facilities with multiple buildings and multiple entrances.
- This method generally requires a security guard to verify that individuals are not hopping over the turnstile or tailgating through an entrance designed for persons with physical disabilities that bypasses the turnstile.
- Turnstiles are not aesthetically pleasing.
- Turnstile access can be inconvenient for employees, escorted guests, or individuals using dollies for equipment.
- This method is more expensive than simple key card access and also has the same issues in the event of a failure in the key card authentication system.

Solving the Single-Factor Identity Problem

A second factor can be added to either of the previous key card authentication processes. The first option is to put a personal identification number (PIN) code reader at every location where there is a card reader. After using their key card, employees must enter a PIN to unlock the door. Another option is to use some form of biometric authentication. Biometric authentication could be used as either the second factor in a key card system or the principal factor in a biometric system. In the second case, users would enter a PIN after successful biometric authentication. See Chapter 4, "Network Security Technologies," for the pros and cons of biometric authentication. Both of these alternatives add cost to the system and inconvenience for users.

Control Physical Access to Data Centers

Data-center access can utilize any of the preceding mechanisms in addition to PIN-reader-only access. The important difference with data-center access is that you are often dealing with a smaller set of operators, so issues around key management are somewhat reduced.

I once had the pleasure of experiencing a physical security audit by a client who was considering using a facility in one of my previous jobs. Needless to say, it didn't go well. One of the auditors was able to gain access to the building by tailgating. Upon entering, he asked to see the "secure" data center we had advertised. Upon reaching the entrance to the secure room, he stood on a chair and pushed up the ceiling tile outside the room. He discovered that the walls to our data center extended only 12 inches beyond the ceiling tiles, allowing access if someone climbed over them.

In the context of this discussion, *data center* refers to any location where centralized network resources are stored. This could include traditional data centers, wiring closets, coat closets, or someone's desk. It all depends on the size of the facility and the way it is organized.

TIP Some ultrasecure data centers utilize sets of cameras, key card access, biometrics, and "man-traps" to catch anyone illegally trying to gain access to the room.

Separate Identity Mechanisms for Insecure Locations

Although identity design considerations are discussed in more detail in Chapter 9, "Identity Design Considerations," from a physical security perspective, it is important to ensure that passwords in physically insecure locations are not the same as those used in secure locations.

Often an organization will utilize common authentication mechanisms for the various systems that must access network resources. For example, SNMP community strings or Telnet/ SSH passwords might be set the same on all devices. From a pure security perspective, it is preferable to use two-factor authentication, when available, for each user who accesses the network device. Although this might be possible for users, it is often impossible for software management systems, which need to run scripts to make changes on several machines at once. For optimal security, different passwords should be used on each device, but this is often operationally impossible for large networks.

Therefore, at a minimum, organize your common passwords so that they are never used on systems in physically insecure locations. For example, assume you have 3 main locations (with data centers) to your organization and 10 remote sites (considered insecure). In this case, only use your shared passwords on the main sites and ensure that the passwords for each of the remote systems are unique per site at a minimum and per device ideally. As the number of insecure locations increases into the hundreds or thousands, this becomes impossible; refer to the "Business Needs" section of Chapter 2, "Security Policy and Operations Life Cycle," for guidance on calculating the costs and benefits of this and any other difficult security measure. (People generally don't compute cost/benefit on easy and cheap security measures.)

Prevent Password Recovery Mechanisms in Insecure Locations

Some devices have controls to prevent the recovery of passwords in the event that an attacker has physical access to your system. For example, on some newer Cisco routers and switches, the command is as follows:

```
Router(config)# no service password-recovery
```

When this command is entered on a router or a switch, interrupting the boot process only allows the user to reset the system to its factory default configuration. Without this command, the attacker could clear the password and have access to the original configuration. This is important because the original configuration might contain common passwords or community strings that would allow the attacker to go after other systems.

This would be particularly useful in insecure branch offices or other locations where the physical security of a network device cannot be assured.

Be Aware of Cable Plant Issues

In today's networks, there are two primary cable types: unshielded twisted pair (UTP) category 5 (or higher) and fiber optic. The risk of an attacker accessing your physical cabling is important to consider because that level of access often can bypass other security controls and provide the attacker with easy access to information (provided encryption is not used). UTP cable is very easy to tap, but it was thought years ago that fiber was immune to cable taps. We now know that this is not the case. The National Security Association (NSA) is rumored to have already tapped intercontinental network links by splicing into the cable; read about it at the following URL: http://zdnet.com.com/2100-11-529826.html.

It is also theorized that fiber cable could be bent far enough so that some light would escape if the outer layer of the cable is removed. With the right types of equipment, this information could then be read.

Additionally, if an attacker gains physical access to a wiring closet or the fiber cable as it runs in a cable tray above a drop ceiling, tapping the cable by installing couplers is another possibility.

All this being said, fiber is more secure than copper because the means to tap the signal are more expensive, difficult to execute, and often require interrupting the original flow of data to install. On the other hand, the means to tap a UTP signal can easily be purchased off of the Internet.

Be Aware of Electromagnetic Radiation

In 1985, the concerns of the paranoid among the security community were confirmed. Wim van Eck released a paper confirming that a well-resourced attacker can read the output of a cathode-ray tube (CRT) computer monitor by measuring the electromagnetic radiation (EMR) produced by the device. This isn't particularly easy to do, but it is by no means impossible. Wim's paper can be found here:

http://www.shmoo.com/tempest/emr.pdf

This form of attack is now commonly called *van Eck phreaking*. Additionally, in 2002, Markus Kuhn at the University of Cambridge published a similar method of reading data off of a CRT, this time by measuring the changes in the amount of light in a room. His paper can be found here:

http://www.cl.cam.ac.uk/~mgk25/ieee02-optical.pdf

And an easy-to-read FAQ on the topic can be found here:

http://www.cl.cam.ac.uk/~mgk25/emsec/optical-faq.html

A simple way to mitigate van Eck phreaking might just be to change the type of font you are using. Ross Anderson and Markus Kuhn did some excellent research on the topic:

http://www.cl.cam.ac.uk/~mgk25/ih98-tempest.pdf

I am certainly not recommending that all systems must address these sorts of security considerations, but it is good to know that such attacks are possible.

Be Aware of Physical PC Security Threats

Oftentimes, inexperienced network designers begin with an unacknowledged assumption that *all* the sensitive data within an organization is contained on servers. In reality, there is sensitive information about my company sitting on the laptop I am using to write this book, as well as on the servers. Like most employees at my company, server resources are used when necessary, but often interesting information is stored locally.

Several physical security issues manifest when you operate under the preceding assumption:

- The first is that portable computer theft is a big problem, not just in the cost of replacing the computer but in the proprietary information that is stored on it. The best protection against having a lost portable computer turn into lost trade secrets is some type of file system encryption. (Some are built into modern OSs.) Chapter 4 has more details on such systems.
- The second is that by compromising the data coming into and out of a PC, you can learn passwords, sensitive data, and so on. An attacker can achieve this through network sniffing, EMR emissions (discussed in the previous section), remote control software (Back Orifice 2000), or novel devices that attach between the keyboard and the PC and record to flash memory every key typed. For more information see this URL:

  http://www.thinkgeek.com/stuff/gadgets/5a05.shtml

Layer 2 Security Considerations

As you learned in Chapter 3, “Secure Networking Threats,” certain attacks run at Layer 2 (L2) of the OSI model. Oftentimes, your posture toward L2 attacks depends on the physical security of the location and the amount of trust you have in users, as defined by your security policy. This section discusses some common design considerations for L2 protocols. The discussion is focused on Ethernet, but most of these issues apply to wireless networks as well.

L2 Control Protocols

Control protocols are usually at the core of any L2 security issue. This section discusses design considerations around L2 control protocol usage. Basic understanding of these protocols is assumed. There are two main topics in this section: the first covers industry-standard protocol considerations; the second covers Cisco-specific protocols.

General Protocol Considerations

This section covers the standard protocols 802.1q, Spanning-Tree Protocol (STP), and briefly mentions 802.1x.

802.1q

The 802.1q standard specifies a standard mechanism for Ethernet switches to exchange virtual LAN (VLAN) information. It adds a 4-byte tag after the source and destination Media Access Control (MAC) addresses. The first 2 bytes act as an Ethernet tag protocol identifier. The second 2 bytes contain all the interesting information. Twelve bits are used as a VLAN identifier (yielding 4096 choices), and 3 bits are used as a priority identifier (in the 802.1p standard). The addition of 4 bytes to the Ethernet packet increases the maximum size of an Ethernet frame from 1518 bytes to 1522 bytes.

When designing a network to take advantage of 802.1q tagging, there are a few security concerns that must be addressed:

- 802.1q has had several implementation flaws in various vendors' equipment over the years. Details of an old Cisco vulnerability can be found here: http://www.sans.org/resources/idfaq/vlan.php. Many of these problems have been fixed, and vendors are beginning to pay more attention to security, particularly as VLANs play a greater role in any network design.
- When using VLANs, the potential for human error increases because the operator must keep track of "virtual" LANs that might not have distinct cable plants associated with them. This can get particularly nasty when you try to remember which VLAN number is the outside of your firewall as opposed to the inside. Good management tools can mitigate the impact of this concern.
- Some attacks that use 802.1q as an attack method are detailed in a later section of this chapter titled "VLAN Hopping Considerations."

STP

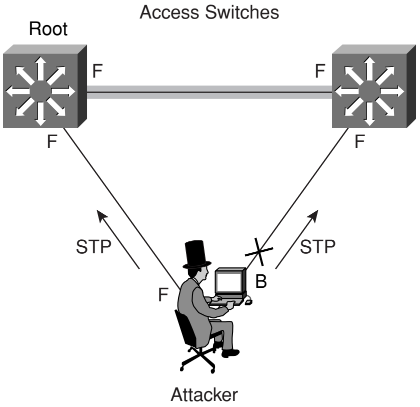
Spanning-Tree Protocol (STP) is a L2 loop avoidance mechanism. Without STP, redundant L2 links would cause large forwarding loops and massive performance problems. From a security standpoint, STP has a few design characteristics of interest.

First, STP has no provisions for authentication of the bridge protocol data units (BPDUs) that are sent from switches and bridges as they exchange STP information. These BPDUs could easily be sent from an unauthorized device that could have any number of undesirable effects.

To start with, if the attacker can cause a failure of a link in the forwarding state, it generally takes 30 to 45 seconds for STP to deal with the failure and reconverge the topology. Some switches now include features to deal with this problem. On Cisco devices, the features are called port fast and uplink fast.

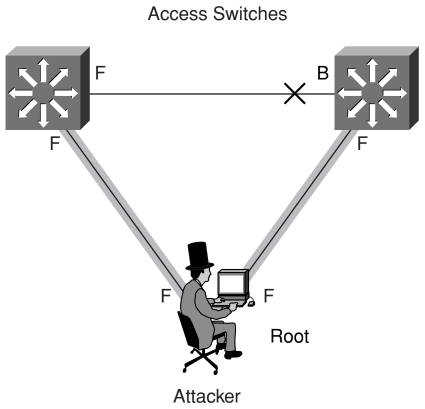
Second, for there to be some "authority" in the STP network, the participating switches elect a *root* bridge. It is from this bridge that the *loop-free* topology is built. The method for determining the root bridge is generally through STP configuration messages, which indicate the bridge priority of a given switch. The lowest number becomes the root bridge. If an attacker is able to send out BPDUs from his station, he can send out a configuration message with a bridge priority of zero. This will likely make his system the root bridge and will often change which links are active on a given network (since the topology is redetermined from the perspective of the new root bridge). No special tools are needed to do this; some UNIX implementations come with Ethernet bridging utilities that allow them to configure their system as a bridge with full participation in the STP process. As an example, consider the following topology in Figure 6-1.

Figure 6-1 *Starting Topology*



In the figure, you can see that the attacker has established two links to two different L2 switches. F denotes a link that is forwarding; B is a link that is blocked because of STP. This could easily be done by walking a long cable to another jack in a building or by using a WLAN network (if it was poorly designed). From here, you can see that one of the attacker's links is in the blocking state. This is exactly what STP should do to prevent loops. However, the attacker then sends BPDUs advertising himself as bridge priority zero. This causes STP to reconverge and the attacker to become the root bridge. A topology that looks like the one in Figure 6-2 results.

Figure 6-2 *Resulting Topology*



Because the topology is built from the perspective of the attacker, you can see that all traffic that must pass between the switches flows through the attacker's PC. This allows an attacker any number of options, as outlined in Chapter 3. The most obvious are sniffing traffic, acting as a man-in-the-middle, or creating a denial of service (DoS) condition on the network. The DoS condition is achieved because the attacker can make his links much slower than the links between the two access switches, which could very likely be connected by gigabit Ethernet.

NOTE You might ask, "Doesn't STP take into account bandwidth speed when determining the topology?" It does but always from the perspective of the root bridge. While testing in the lab, I was able to take a full-duplex gigabit link between two access switches and reduce it to a half-duplex 10 megabit (Mb) connection between those access switches and the attacking PC. This is never good for a production network.

Fortunately, mitigating this attack is fairly straightforward. First, some advocate disabling STP in all cases in which you don't have network loops. Although this sounds like a good idea, the attacker could instead introduce a loop into your network as a means of attack. A better option is to filter which ports are allowed to participate in the STP process. Some switches offer the ability to do this today. On Cisco devices, the two principal options are BPDU Guard and Root Guard.

BPDU Guard BPDU Guard can be globally enabled on some Cisco switches and is in effect on any port configured with the port fast option. Port fast ports are generally user ports. What BPDU Guard does is disable any port fast port that receives a BPDU message. Because these are user ports, there should be no reason for BPDU messages to be sent to them. The syntax is as follows:

```
CatOS> (enable)set spantree portfast bpdu-guard enable
IOS(config)#spanning-tree portfast bpduguard
```

Root Guard The other option you have is Root Guard. Root Guard can be enabled or disabled on any port and works by disabling a port that would become the root bridge as a result of its BPDU advertisement. This is less restrictive on users because it allows them to plug in an Ethernet switch in their workspace (in case they have more than one PC). The syntax for Root Guard is as follows:

```
CatOS> (enable) set spantree guard root 1/1
IOS(config-if)#spanning-tree guard root (or rootguard)
```

I learned about BPDU Guard the hard way. I was setting up a small lab in my office to do some testing, and I needed more ports than I had available. I plugged in an Ethernet switch and promptly lost link on my connection. Puzzled, I went to the IT staff, who informed me that BPDU Guard was running to prevent unauthorized STP advertisements. After getting my port reset, I went back to my office and turned off STP on my small switch. Problem solved.

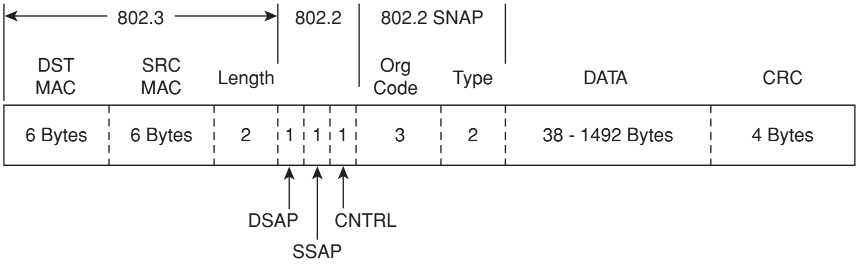
802.1x

The standard 802.1x specifies a mechanism to do port-based access control in an Ethernet network. For example, before granting access to a user who connected to a port in one of your conference rooms, you could have 802.1x require authentication first. Upon authentication, the user could be assigned to a specific VLAN based on the user's access rights. The 802.1x standard could be used in the future to perform additional security checks, perhaps enforcing an access control list (ACL) for the user or a quality of service (QoS) policy. The 802.1x standard is covered further in Chapter 9.

Cisco-Specific Protocols

Over the years, Cisco Systems has developed a number of proprietary protocols that have been used to perform different functions on an L2 network. Most of these protocols use an IEEE 802.3 frame format with an 802.2 SNAP encapsulation. Most have a Logical Link Control (LLC) of 0xAAAA03 (indicating SNAP) and the Cisco Organizational Unit Identifier (OUI) 0x00000c. The majority use a multicast destination MAC address to communicate. This is generally a variation on 0100.0ccc.cccc. The SNAP protocol type varies and generally is included in each protocol discussion where appropriate. Knowing the specifics of these protocols should make it easier to identify them on the network when troubleshooting using a sniffer. Figure 6-3 shows in detail the frame format of most Cisco L2 protocols.

Figure 6-3 *802.3 with 802.2 SNAP Frame Format*



- DST MAC: Generally a variant of 0100.0ccc.cccc
- SRC MAC: Pulled from a pool in the switch EPROM
- 802.2 LLC Fields
  DSAP:AA + SSAP:AA + CNTRL:03 = SNAP
- 802.2 SNAP Fields
  Org Code:0x00000c (Cisco)
  Protocol Type: Varies

Of special note is the unique relationship two of these protocols have with VLAN 1 on Cisco switches. Cisco Discovery Protocol (CDP) and VLAN trunking protocol (VTP) are discussed in more detail later. Both of these protocols communicate over VLAN 1 only. Even if VLAN 1 is not used on a trunk port, these protocols continue to pass information on VLAN 1. For this reason, and the fact that VLAN 1 cannot be deleted, it is not recommended to use VLAN 1 for user or trunk ports. More information on this topic can be found here: http://www.cisco.com/warp/public/473/103.html.

Interswitch Linking (ISL)

Long before 802.1q, Cisco switches were capable of trunking multiple VLANs over a single link using ISL. ISL use is in decline because there is now an adequate standard to replace it. Instead of using a 4-byte field in the Ethernet frame, ISL reencapsulates the packet with a new Ethernet header, adding 26 bytes to the packet (10 bits is used for the VLAN ID). If you remember, 802.1q adds only 4 bytes to each packet (including priority by 802.1p) and, as such, is a more efficient protocol. Although it is not recommended to build a new network from scratch using ISL, many existing networks run ISL. The security issues around ISL are virtually identical to those of 802.1q.

Dynamic Trunking Protocol (DTP)

To help switches determine whether they should be trunking, Cisco developed DTP. DTP exchanges information between switches, notifying each other of their preferences regarding trunking for a given link. Settings such as auto, on, off, desirable, and non-negotiate determine whether a given L2 switch will trunk on a given link. DTP uses a destination MAC address of 0100.0ccc.cccc and a SNAP protocol type of 0x2004. Cisco Catalyst 2900XL and 3500XL switches do not support DTP.

DTP is important from a security perspective because the default DTP state of many switches is auto. This means that they will happily trunk (pass traffic on multiple VLANs) with anyone who notifies them that they would like to do so. DTP spoofing is a part of the attacks described in the "VLAN Hopping Considerations" section later in this chapter. To mitigate attacks that use DTP, it is recommended that you set all ports without a need to trunk into the DTP off state. The syntax for these commands is as follows:

```
CatOS> (enable) set trunk <mod/port> off
IOS(config-if)#switchport mode access
```

If you aren't sure whether your switch defaults to autotrunking or not, you can check the trunk status of your ports with the following commands:

```
CatOS> (enable) show trunk [mod¦mod/port]
IOS#show interface type number switchport
```

VLAN Trunking Protocol (VTP)

Oftentimes, it can be a burden to manage a large L2 network with lots of VLANs spread around different switches. To ease this burden, Cisco developed VTP. VTP allows an administrator to configure a VLAN in one location and have its properties automatically propagated to other switches inside the VTP domain. VTP uses a destination MAC address of 0100.0ccc.cccc and a SNAP protocol type of 0x2003. VTP uses the notion of a client and a server to determine which devices have rights to propagate VLAN information in what direction.

I'll be honest, having my VLAN information automatically propagate to my different switches doesn't fill the security part of my brain with glee. Start by strongly considering whether VTP is going to save you time or cause you headaches. If all your VLANs have similar security levels, perhaps VTP could be helpful to you. But if instead you have different security levels on your VLANs and certain VLANs should only exist on certain switches, it is probably easier, and safer, to manually configure each VLAN where you need it.

If you must use VTP, be sure to use it with the MD5 digest option. This adds a 16-byte MD5 digest of the VTP packet combined with a password and makes it much harder for an attacker to send you bogus VTP information causing your VLANs to be reconfigured. Without the MD5 authentication, an attacker could be disguised as a VTP server with all VLANs deleted. This could cause all switches in your entire network to remove their VLAN configuration. Not a good thing for security at all! The syntax for configuring a VTP password is as follows:

```
CatOS> (enable) set vtp [domain domain_name]
[mode {client ¦ server ¦ transparent ¦ off}] [passwd passwd]
[pruning {enable ¦ disable}] [v2 {enable ¦ disable}]
IOS(config)#vtp password password-value
```

VLAN Query Protocol (VQP)

Prior to the establishment of the IEEE 802.1x standard, Cisco developed a technology called the VLAN Management Policy Server (VMPS). VMPS works with a flat file policy database that is sent to VMPS server switches by TFTP. VMPS client switches then communicate with the VMPS server using VQP. VMPS allows a switch to dynamically assign VLANs to users based on their MAC address or user identity (if used with the User Registration Tool [URT]).

Unfortunately, VQP is a UDP-based protocol that does not support any form of authentication. This makes its use in security-sensitive environments inadvisable. An attacker who is able to spoof VQP (not hard since it runs UDP) could then try to prevent network logins or might join a VLAN unauthorized.

VQP and VMPS are rarely used for MAC-based VLAN assignment because of the management burden of maintaining the MAC address to VLAN mapping table. The URT component is also not frequently used, especially since a standards-based method of effectively doing the same thing (802.1x) is now available.

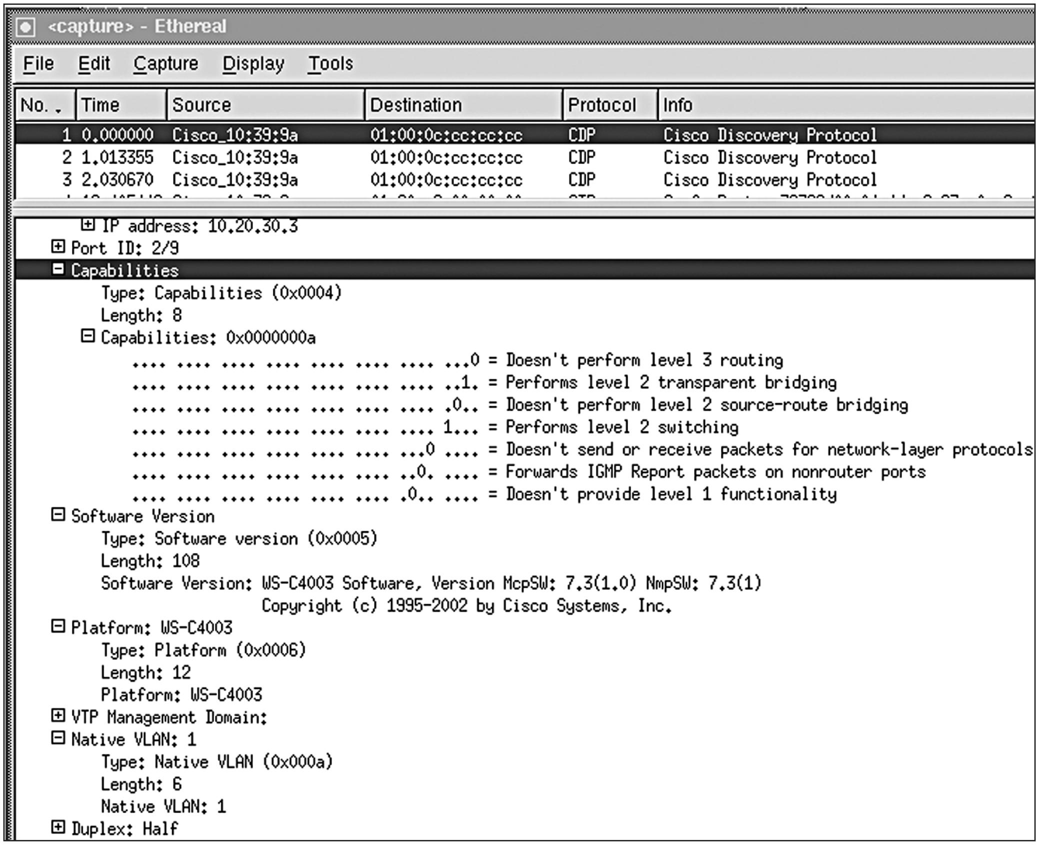
CDP

To allow Cisco devices to exchange information about one another's capabilities, Cisco developed CDP. CDP uses a destination MAC address of 0100.0ccc.cccc and a SNAP protocol type of 0x2000. By default, most Cisco routers and switches have CDP enabled. CDP information is sent in periodic broadcasts that are updated locally in each device's CDP database. Because CDP is an L2-only protocol, it (like any other L2 protocol discussed here) is not propagated by routers. Some of the types of data propagated by CDP include the following:

- L2/L3 capabilities
- Hostname
- Native VLAN
- Duplex setting
- Software version
- VTP domain settings

Figure 6-4 is a portion of an Ethereal packet trace showing the inside of a CDP packet.

Figure 6-4 *CDP Example Packet*



```
No.  Time      Source            Destination        Protocol  Info
  1 0.000000  Cisco_10:39:9a    01:00:0c:cc:cc:cc  CDP       Cisco Discovery Protocol
  2 1.013355  Cisco_10:39:9a    01:00:0c:cc:cc:cc  CDP       Cisco Discovery Protocol
  3 2.030670  Cisco_10:39:9a    01:00:0c:cc:cc:cc  CDP       Cisco Discovery Protocol
```

```
        ⊞ IP address: 10.20.30.3
    ⊞ Port ID: 2/9
    ⊟ Capabilities
        Type: Capabilities (0x0004)
        Length: 8
      ⊟ Capabilities: 0x0000000a
            .... .... .... .... .... .... .... ...0 = Doesn't perform level 3 routing
            .... .... .... .... .... .... .... ..1. = Performs level 2 transparent bridging
            .... .... .... .... .... .... .... .0.. = Doesn't perform level 2 source-route bridging
            .... .... .... .... .... .... .... 1... = Performs level 2 switching
            .... .... .... .... .... .... ...0 .... = Doesn't send or receive packets for network-layer protocols
            .... .... .... .... .... .... ..0. .... = Forwards IGMP Report packets on nonrouter ports
            .... .... .... .... .... .... .0.. .... = Doesn't provide level 1 functionality
    ⊟ Software Version
        Type: Software version (0x0005)
        Length: 108
        Software Version: WS-C4003 Software, Version McpSW: 7.3(1.0) NmpSW: 7.3(1)
                          Copyright (c) 1995-2002 by Cisco Systems, Inc.
    ⊟ Platform: WS-C4003
        Type: Platform (0x0006)
        Length: 12
        Platform: WS-C4003
    ⊞ VTP Management Domain:
    ⊟ Native VLAN: 1
        Type: Native VLAN (0x000a)
        Length: 6
        Native VLAN: 1
    ⊞ Duplex: Half
```

From a reconnaissance standpoint, all of the preceding information could be useful to an attacker. The software version, in particular, would allow the attacker to determine whether there were any specific security vulnerabilities with that particular version of code. Also, since CDP is unauthenticated, an attacker could craft bogus CDP packets and have them received by the attacker's directly connected Cisco device.

So, with an understanding of the security risks, why don't you just turn CDP off completely? Many network operators do, but it is important to realize that Cisco developed CDP for a reason. Some network management applications make use of it, as do Cisco IP telephones. If you must run CDP on your network, consider using it on only the ports that require its use. For example, many networks need CDP only on backbone links and not user links. This would allow you to turn off CDP on user ports, preventing many of the attacks discussed in the preceding paragraph. The syntax to disable CDP on a router or a switch is as follows:

```
CatOS> (enable) set cdp disable <mod>/<port> ¦ all
IOS(config)#no cdp run
IOS(config-if)#no cdp enable
```

MAC Flooding Considerations

Every L2 switch needs some mechanism to record the port to which a given MAC address is connected. This ensures that unicast communication between two hosts can occur without other hosts seeing the traffic. One common method of recording this information is the use of a Content Addressable Memory (CAM) table. A CAM table stores the MAC addresses and VLAN assignments of various hosts connected on a switch. Think of it much like a routing table for a router, only at L2.

When a frame arrives at a switch, a number of things happen. The sequence we refer to here is specific to the CAM table and frame switching:

1. The frame arrives at the switch.
2. The source MAC address is inspected to determine whether there is already an existing entry in the CAM table. If so, the switch proceeds to the next step; if not, an entry is added to the CAM table for the source MAC address. This way, when anyone needs to talk to that MAC address again, the switch remembers which port to send the frame to reach the destination.
3. The destination MAC address is inspected to determine whether there is already an existing entry in the CAM table. If so, the frame is switched out of that destination port and on to the host. If not, the switch proceeds to step 4.
4. The switch floods the frame on all ports that are members of the same VLAN as the originating host.
5. When the intended recipient of the frame receives the packet, it responds (assuming the protocol is two-way), and the switch repeats this process from step 1. The switch adds an entry in the CAM table for the source MAC of this frame (the destination MAC of the previous frame). All further unicast communications between these two hosts are sent on only the port to which each host is connected.

The preceding illustrates how it is supposed to work. A security-conscious network designer must be aware of a few things:

- CAM tables have a limited size. Depending on the switch, this can be anywhere from 100 or so entries to over 100,000 entries.
- Entries in the CAM table have an aging timer. Each time a frame is transmitted with a source MAC address matching the current entry in the CAM table, the aging timer is reset. If a given host does not send frames on the switched network, the network eventually

deletes the CAM table entry for that device. This is of particular interest for one-way protocols such as syslog. If your syslog server does nothing but receive UDP syslog messages, its CAM entry will begin to age once it responds to the original Address Resolution Protocol (ARP) query sent by a host or router. Once aged, all packets destined for it are always flooded on the local VLAN.

Attack Details

Given the previous explanation of how a CAM table works, let's look at how the CAM table design can be attacked:

1. An attacker connects to a switch port.
2. The attacker sends a continuous set of frames with random source MAC addresses and random destination MAC addresses. The attacker is really concerned with making sure steps 1 and 2 of the preceding list repeat constantly, each time with a different MAC address.
3. Because CAM tables have limited size, eventually the switch will run out of room and not have any more space for new MAC addresses.
4. A victim host (connected to the same VLAN as the attacker) tries to communicate with a host that does not currently have a CAM table entry.
5. Since there is no more room in the CAM table for the host without an entry, all communications to that host must be flooded.
6. The attacker can now see all the traffic sent from the victim host to the host without a CAM table entry. This could include passwords, usernames, and so forth, which then allows the attacker to launch the next attack.

This attack is important because Ethernet switches were originally thought to increase security because only the ports involved in a particular communication would see the traffic. Furthermore, if the attacker runs this attack continuously, even active hosts might soon start flooding as the aging timer expires during periods of inactivity. The attacker can further accelerate this process by sending an STP BPDU with a Topology Change Notification (TCN) message (such as when an attacker tries to become the root bridge). Such a message will cause the aging timer on most switches to temporarily shorten. This is needed so the switch doesn't keep stale information that is no longer valid after the STP topology change. For example, many Cisco switches have a default aging timer of 300 seconds. When a Cisco switch receives a TCN message, it automatically reduces the aging timer for every entry to 15 seconds.

As mentioned in Chapter 3, there are several popular tools that automate this attack. The most common is macof, written in 1999 by Ian Vitek in about 100 lines of Perl. This code was later ported to C by Dug Song for his dsniff tools. In a very basic lab test, I was able to generate 155,000 MAC entries per minute using a stock Linux box.

There are a few caveats to this attack that you should be aware of:

- Even with a completely full CAM table, traffic is flooded only on the local VLAN, meaning traffic on VLAN 10 stays on VLAN 10, but everyone with a port on VLAN 10 will see the traffic.
- Because of the flooding, this attack could also flood the CAM table on adjacent switches.
- Because of the sheer quantity of traffic the attacker sends, this attack might also result in a DoS condition on the network.

Attack Mitigation

Stopping this attack isn't too difficult, but it isn't quite as simple as flipping a switch. Many switches offer the ability to do something called *port security*. Port security works by limiting the number of MAC addresses that can communicate on any given port on a switch. For example, say you are running switched Ethernet to the desktop in your environment. Each host has its own connection on the switch. Here, you might configure port security to allow only one MAC address per port. Just to be safe, you might allow two or three in case locations add a small hub to connect a test system. Port security works by learning the number of MAC addresses it is configured to allow per port and then shutting down the port if it exceeds the limit. In the case of the macof tool, it would be stopped dead in its tracks. You configure port security in the following way:

```
CatOS> (enable)   set port security mod/ports... [enable ¦ disable] [mac_addr]
 [age {age_time}] [maximum {num_ of_mac}] [shutdown {shutdown_time}]
[violation{shutdown ¦ restrict}]
IOS(config-if)#port security [action {shutdown ¦ trap} ¦ max-mac-count addresses]
```

Note that there are a lot of other options that aren't really necessary for stopping CAM table flooding. For more information on port security, you can look here: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_4/config/sec_port.htm.

For example, here's a configuration in Cisco CatOS to limit ports to two MAC addresses:

```
CatOS> (enable)   set port security 3/1-48 enable maximum 2
```

This uses the default of a permanent shutdown in the event of a violation. There are other options, such as setting a timer on how long the port is shut off or deciding instead to leave the port operational but drop any MAC addresses that aren't in the original set allowed by the switch. This latter option is inadvisable because it can create increased load on the switch while it tries to determine which traffic to pass or drop. It is also worth noting that this attack, like all L2 attacks, requires the attacker to have local access to the network because these attacks do not cross a router.

VLAN Hopping Considerations

Since VLANs were first created, there has been debate over their use in a security role. The threat of VLAN hopping (causing traffic from one VLAN to be seen by another VLAN without first crossing a router) was and is still viewed as the major risk. Designers want to know whether it is safe to design their networks as shown in Figure 6-5 instead of using additional switches as shown in Figure 6-6.

Figure 6-5 *Questionable VLAN Edge Design*

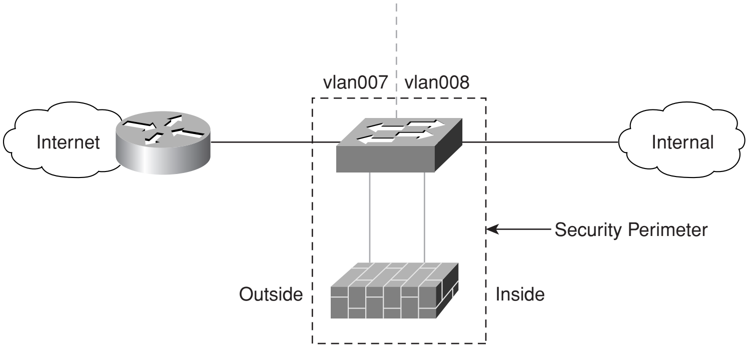
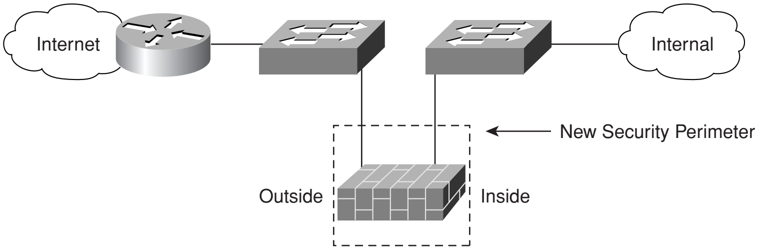


Figure 6-6 *Edge Design without VLANs*



The short answer is, assuming your Ethernet switch vendor doesn't have any security-related bugs with VLANs, VLANs can be deployed in a reasonably secure manner. Unfortunately, the precondition of no bugs is a hard state to achieve. A number of bugs have allowed VLAN hopping over the years. The best you can hope for is that any bugs that are discovered with VLAN security are quickly fixed by your vendor. Additionally, misconfigurations can sometimes allow VLAN hopping to occur, as you'll see in the following two sections.

Basic VLAN Hopping Attack

In the basic VLAN hopping attack, the adversary takes advantage of the default configuration on most switches. As we discussed in the preceding section on DTP, most switch ports default to autotrunking. This means that an attacker that can successfully trick a switch into thinking it is another switch with a need to trunk can gain access to all the VLANs allowed on the trunk port. This can be achieved in one of two ways:

- Spoof the DTP messages from the attacking host to cause the switch to enter trunking mode. From here, the attacker can send traffic tagged with the target VLAN, and the switch will happily deliver the packets to the destination.
- Introduce a rogue switch and turn trunking on. The attacker can then access all the VLANs on the victim switch from the rogue switch.

This basic VLAN hopping attack can be easily mitigated by turning trunking off on all ports without a specific need to trunk. The configuration settings for this are shown in the DTP section earlier in this chapter.

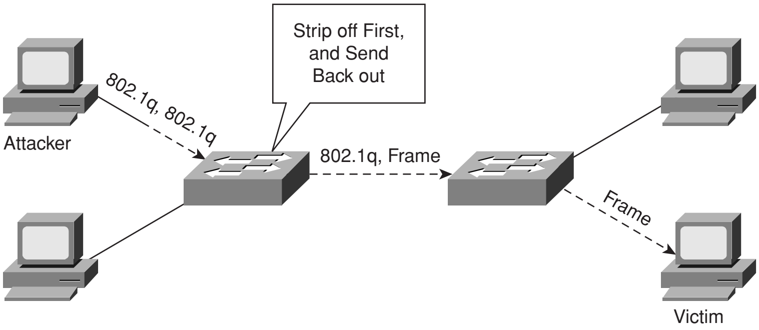
Creative VLAN Hopping Attacks

This section is a catchall for various methods to achieve VLAN hopping when trunking is turned off on the port to which the attacker is connected. As these methods are discovered, they tend to be closed by the vendors affected. One tricky attack will take some time to stop on all devices. You might wish to refer to the previous section on 802.1q if you need more information. The attack works by sending frames with two 802.1q tags instead of one. The attack requires the use of two switches, and the attacker and victim must be on separate switches. In addition, the attacker and the trunk port must have the same 802.1q native VLAN. The attack works like this:

1. The attacker sends a double-tagged 802.1q frame to the switch. The outer header has the VLAN tag of the attacker and trunk port. (For the purposes of this attack, let's assume VLAN 10.) The inner tag is the victim VLAN, VLAN 20.
2. The frame arrives on the switch, which looks at the first 4-byte 802.1q tag. The switch sees that the frame is destined for VLAN 10 and sends it out on all VLAN 10 ports (including the trunk) since there is no CAM table entry. Remember that, at this point, the second VLAN tag is still intact and was never inspected by the first switch.
3. The frame arrives at the second switch but has no knowledge that it was supposed to be for VLAN 10. (Remember, native VLAN traffic is not tagged by the sending switch as specified in the 802.1q spec.)
4. The second switch looks at only the 802.1q tag (the former inner tag that the attacker sent) and sees the frame is destined for VLAN 20 (the victim VLAN).
5. The second switch sends the packet on to the victim port or floods it, depending on whether there is an existing CAM table entry for the victim host.

Figure 6-7 illustrates the attack. It is important to note that this attack is only unidirectional and works only when the attacker and trunk port have the same native VLAN.

Figure 6-7 *Double-Tagged 802.1q VLAN Hopping Attack*



This attack is easy to stop if you follow the best practice that native VLANs for trunk ports should never be used anywhere else on the switch. For switches to prevent this attack, they must look further into the packet to determine whether more than one VLAN tag is attached to a given frame.

Unfortunately, the application-specific integrated circuits (ASICs) that are used by most switches are only hardware optimized to look for one tag and then to switch the frame. The problem of performance versus security rears its ugly head again.

TIP You might be wondering why the switch is accepting tagged frames on a port that isn't trunking in the first place. Refer to the section on 802.1q, where we discussed that part of the 802.1q tag is the 802.1p tag for frame priority (QoS). So, to support 802.1p, the switch must support 802.1q frames.

ARP Considerations

ARP is designed to map IP addresses to MAC addresses. It was also, like most protocols still used in IP networking today, designed at a time when everyone on a network was supposed to be reasonably trustworthy. As a result, the protocol is designed around efficiently executing its task, with no provisions for dealing with malicious use. At a basic level, the protocol works by broadcasting a packet requesting the MAC address that owns a particular IP address. All devices on a LAN will see the request, but only the device that uses the IP address will respond.

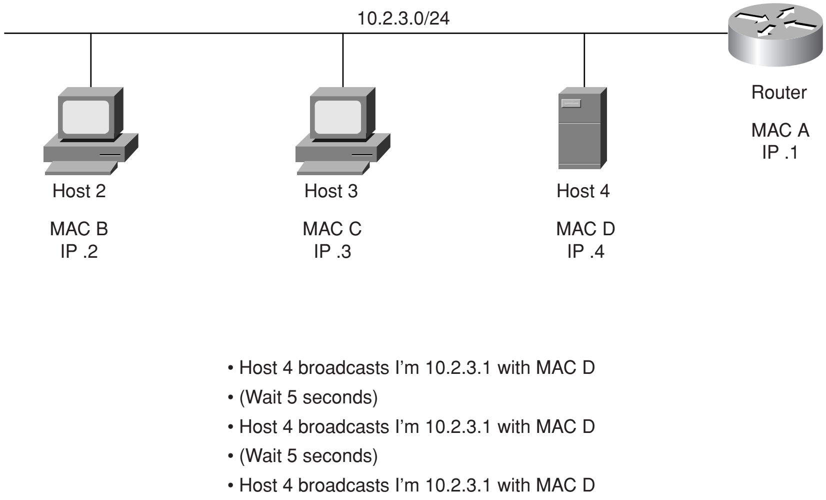
From a security standpoint, there is a major limitation in ARP. ARP has no notion of IP address ownership. This means any MAC address can masquerade as any IP address provided an attacker has the right software tool to execute the attack. Furthermore, there is a special type of

ARP broadcast called a gratuitous ARP (gARP). A gARP message tells all hosts on a LAN, without having been asked, what its IP–MAC binding is.

gARP is used in several legitimate ways. The most prevalent is in high-availability situations in which two systems share the same IP address but have different MAC addresses. When the primary system changes, it must notify the rest of the LAN of the new MAC address with which to contact the primary host. ARP is also used to prevent IP address conflicts. Most modern OSs send an ARP request out for the address with which they are configured when they boot. If a machine responds, they know that another node is already using their configured IP address, and the interface should be shut down until the conflict can be resolved.

Consider the following sequence outlined in Figure 6-8.

Figure 6-8 *Misuse of gARP*



- Host 4 broadcasts I’m 10.2.3.1 with MAC D
- (Wait 5 seconds)
- Host 4 broadcasts I’m 10.2.3.1 with MAC D
- (Wait 5 seconds)
- Host 4 broadcasts I’m 10.2.3.1 with MAC D

In the figure, a host that is not the router is sending gARP broadcasts claiming to be the router’s IP address but using its own MAC address. Hosts 2 and 3 generally ignore such a broadcast if they haven’t yet communicated with the router. When they finally do, they send an ARP request for the router’s MAC address. The real router (.1) will respond, but as soon as host 4 sends the next gARP broadcast claiming to be .1, hosts 2 and 3 will update their ARP entry for .1 to reflect host 4’s MAC address (MAC D).

At this point, the traffic destined off of the 10.2.3.0/24 network will go to host 4’s MAC address. That host could then send it to the real router, drop the traffic, sniff the traffic, or modify the contents of a packet and send it along to the real router.

Then all traffic from the hosts flows through the attacker's machine before arriving at the actual router. If desired, the attacker could also send gARP broadcasts to the router claiming to be every host on the local LAN, which allows the attacker to see the return traffic as well.

The attack described in the preceding paragraphs is the core problem with ARP. The attack described is generally referred to as ARP redirection or spoofing. Any host on the LAN can attempt to masquerade as any other host through the use of ARP and gARP messages.

dsniff is a collection of tools written by Dug Song to launch and further take advantage of this attack. For example, after launching the ARP spoofing attack, dsniff has a special sniffer designed to find and output to a file the usernames and passwords of dozens of common protocols. It even goes so far as to execute man-in-the-middle (MITM) attacks against Secure Sockets Layer (SSL) and SSH by presenting false credentials to the user. By using this attack, it becomes possible for an attacker to learn sensitive information sent over encrypted channels. More information on dsniff can be found at the dsniff website: http://monkey.org/~dugsong/dsniff/.

Mitigating ARP redirection attacks is a bit trickier. You could use private VLANs (PVLANs) as described later in this section, but this would prevent all host-to-host communication, which isn't particularly good for a network (except in specific cases such as server farms). A feature available in some Cisco switches is called ARP inspection. ARP inspection allows VLAN ACLs (VACLs) to be applied to ARP traffic flowing across a specific VLAN on the switch. A common way these VACLs are used is to make sure the MAC address of the default gateway does not change. The following ACL restricts ARP messages for two MAC–IP bindings and prevents any other MAC address from claiming ownership for those two IPs:

```
CatOS> (enable) set security acl ip ACL-95 permit arp-inspection host
192.0.2.1 00-d0-b7-11-13-14
CatOS> (enable) set security acl ip ACL-95 deny arp-inspection host
192.0.2.1 any log
CatOS> (enable) set security acl ip ACL-95 permit arp-inspection host
192.0.2.2 00-d0-00-ea-43-fc
CatOS> (enable) set security acl ip ACL-95 deny arp-inspection host
192.0.2.2 any log
CatOS> (enable) set security acl ip ACL-95 permit arp-inspection any any
CatOS> (enable) set security acl ip ACL-95 permit ip any any
CatOS> (enable) commit security acl ACL-95
```

As you can see, you must first permit the explicit binding. Then you deny any other ARP packets for that same IP. Finally, you permit all other ARP packets.

There are some caveats to ARP inspection as it is currently implemented, and the management burden of tracking MAC address and IP bindings for ACL entries probably prevents many system administrators from using this for anything other than default gateways and critical systems. For more information on ARP inspection, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_7_5/confg_gd/acc_list.htm#1020673.

You can also limit on a per-port basis the number of ARP packets that are processed by the switch. Excess packets are dropped and can optionally cause the port to shut down. This can stop really noisy ARP attacks, but most ARP tools are less noisy than this. Arpspoof, for example, sends less than one ARP message per second. The following example sets an inspection limit of 25 packets per second and a shutdown threshold of 50 packets per second for port 2/1.

```
CatOS> (enable) set port arp-inspection 2/1 drop-threshold 25 shutdown-
 threshold 50
Drop Threshold=25, Shutdown Threshold=50 set on port 2/1.
CatOS> (enable)
CatOS> (enable) show port arp-inspection 3/1
Port Drop Threshold Shutdown Threshold
------------------------ -------------- ------------------
2/1                 25       50
```

Keep in mind that, when systems initialize, they might send large numbers of legitimate ARP queries. Use this feature with caution, especially considering it won't stop the ARP attacks used today. If you deploy ARP inspection, be sure to use the VACLs as your primary means of defense and the ARP rate limiting to stop clearly nonstandard behavior.

Other methods that can help include hard-coding static ARP entries for key devices in your network. From a management standpoint, you'd never be able to do this for all hosts, but for key devices it might be worth the effort.

TIP Unfortunately, some older Microsoft operating systems (OSs) allow a static ARP entry to be overwritten by a gARP broadcast.

Open source tools can be used to help as well: arpwatch is a free tool developed by Lawrence Berkeley National Lab (LBNL). It works by keeping track of IP and MAC address bindings on the network and can notify you when certain mappings change. The tool can be downloaded here: http://www-nrg.ee.lbl.gov/.

Last, some IDS tools have the ability to detect certain types of ARP attacks. Some look for large quantities of ARP traffic, while others operate in much the same way as arpwatch.

DHCP Considerations

Dynamic Host Configuration Protocol (DHCP) allows hosts to request IP addresses from a central server. Additional parameters are usually passed as well, including DNS server IP address and the default gateway.

DHCP can be attacked in two ways:

- Attackers could continue to request IP addresses from a DHCP server by changing their source MAC addresses in much the same way as is done in a CAM table flooding attack. A tool to execute such an attack is available here: http://packetstormsecurity.org/DoS/DHCP_Gobbler.tar.gz. If successful, the attack will cause all the leases on the DHCP server to be allocated.
- The second attack is a bit nastier. Here, the attacker introduces a rogue DHCP server into the network. The server then attempts to offer DHCP addresses to whomever requests them. The fields for the default gateway and DNS server are set to the attacker's host, enabling all sorts of sniffing and MITM attacks much like dsniff. Even if your real DHCP server is operational, it doesn't mean you won't get a rogue address. What happens to you depends on the host OS you are running. Here is the relevant bit from the DHCP RFC 2131:

  > The client collects DHCPOFFER messages over a period of time, selects one DHCPOFFER message from the (possibly many) incoming DHCPOFFER messages (e.g., the first DHCPOFFER message or the DHCPOFFER message from the previously used server) and extracts the server address from the "server identifier" option in the DHCPOFFER message. The time over which the client collects messages and the mechanism used to select one DHCPOFFER are implementation dependent.

I tested a number of different OSs and all accepted the first DHCP offer they received, whether it was for their old IP address or not.

The method used to stop the first attack is identical to how you stop the CAM table flooding attack: use port security. The second attack is more difficult to stop. DHCP Authentication (RFC 3183) will help but has not yet been implemented (and also has some nasty key management implications). Both DHCP snooping and specific VACLs can help and are defined in the next sections.

DHCP Snooping

Some Cisco switches offer the ability to suppress certain types of DHCP information on certain ports. The primary feature enabling this functionality is DHCP snooping. DHCP snooping works by separating trusted from untrusted interfaces on a switch. Trusted interfaces are allowed to respond to DHCP requests; untrusted interfaces are not. The switch keeps track of the untrusted port's DHCP bindings and rate limits the DHCP messages to a certain speed. The first task in configuring DHCP snooping is to enable it:

```
Switch(config)#ip dhcp snooping
```

From here, DHCP snooping must be enabled for specific VLANs:

```
Switch(config)#ip dhcp snooping vlan number [number]
```

WARNING As soon as you enter the VLAN-specific DHCP command, all DHCP stops working until you trust the ports for the DHCP server with the DHCP snooping **trust** command. You should enter the **trust** command first if deploying to a production network.

To set up the trusted ports at the interface level, ports must be defined as trusted or untrusted using the following command:

```
Switch(config-if)# ip dhcp snooping trust
```

Untrusted ports can be optionally configured with a rate limit on the amount of DHCP messages allowed per second:

```
Switch(config-if)# ip dhcp snooping limit rate rate
```

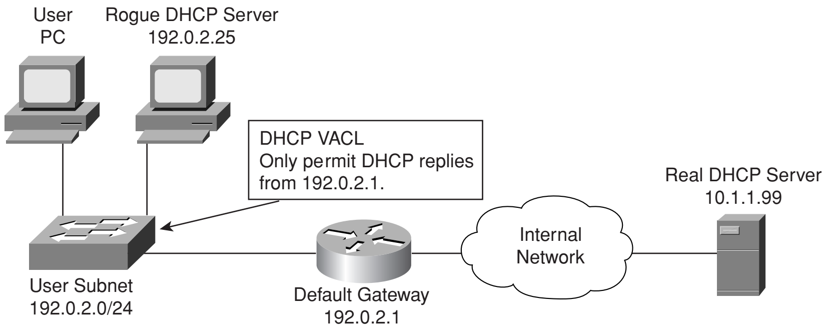
WARNING Do not enable rate limiting on a trusted port because, when the rate is exceeded, the port is shut down. Rate limiting is designed more to protect the DHCP snooping process on the switch than to stop any DHCP attacks. Most DHCP attacks have a very low packet per second (pps) count.

DHCP snooping is not particularly useful if there are multiple systems behind a port on a switch (through either a hub or another switch). In these environments, the rouge DHCP server could sit off of this switch or hub and attack the local systems. For more information on other options for DHCP snooping, see the following: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_1_13/config/dhcp.htm.

DHCP VACLs

Not all switch deployments are able to take advantage of DHCP snooping. A lower-tech solution to this problem can be partially achieved with DHCP VACLs. The VACL can specify which addresses are able to send DHCP replies. These replies will come from the unicast IP address of the DHCP server offering the lease. By filtering these replies by source address, rogue DHCP servers can be properly filtered. Consider the typical DHCP deployment depicted in Figure 6-9.

Figure 6-9 *Common DHCP Deployment*



Here, a local LAN is being served by a remote DHCP server. This server receives DHCP requests by DHCP relay configured on the default router. When the default router receives the DHCP lease offer back from the DHCP server, it passes it on to the client directly. Here is a VACL to protect against rogue DHCP servers in this example:

```
set security acl ip ROGUE-DHCP permit udp host 192.0.2.1 any eq 68
set security acl ip ROGUE-DHCP permit udp host 10.1.1.99 any eq 68
set security acl ip ROGUE-DHCP deny udp any any eq 68
set security acl ip ROGUE-DHCP permit ip any any
```

From the point at which the user PC requests an initial lease, here is what happens:

1. The user PC boots up and sends a DHCP request with source 0.0.0.0 and destination 255.255.255.255.
2. Both the default router and the rogue DHCP server see this request.
3. The rogue DHCP server replies, but since the source IP address is not 192.0.2.1, the reply is dropped by the access switch.
4. The default router passes the DHCP request to the real DHCP server, receives a reply, and passes this information on to the client.
5. The client connects and uses the network.

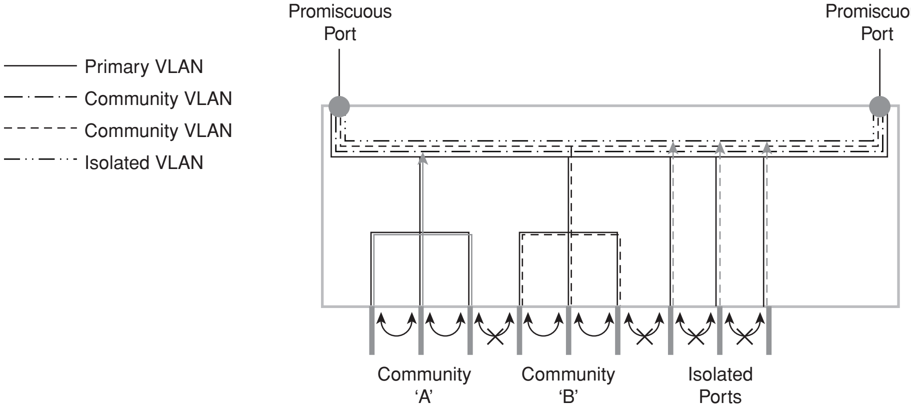
WARNING Using VACLs to stop rogue DHCP servers is far from comprehensive protection. The rogue server could still spoof the IP address of the legitimate DHCP server. However, using VACLs will certainly stop all accidental DHCP servers put on the network and will thwart most common attackers.

Private VLANs

PVLANs offer further subdivision within an existing VLAN, allowing individual ports to be separated from others while still sharing the same IP subnet. This allows separation between devices to occur without requiring a separate IP subnet for each device (and the associated IP addresses that would waste). In its simplest form, PVLANs support isolated ports and promiscuous ports. Isolated ports can talk only to promiscuous ports, while promiscuous ports can talk to any port. In this deployment, the members of a subnet are isolated ports, and the gateway device is connected to a promiscuous port. This enables the hosts on a subnet to offer services to other subnets and to initiate requests of other subnets but not to service the requests of members of the same subnet.

A further PVLAN option available on some switches is community ports. In this model several isolated ports can be considered part of a community, enabling them to communicate with each other and the promiscuous port but not with other communities or isolated ports. Figure 6-10 summarizes these options.

Figure 6-10 *PVLANs*



The most common security-related deployment of PVLANs is in a public services segment or demilitarized zone (DMZ) connected to a firewall. In this deployment, PVLANs prevent the compromise of one system from leading to the compromise of other systems connected to the same subnet. Without PVLANs, an attacker could go after other vulnerable systems on any port or protocol because the attacker is already past the firewall. For example, a server segment off of your main corporate firewall might have FTP, SMTP, and WWW servers. There probably isn't much need for these devices to communicate with one another, so PVLANs can be used.

Configuring PVLANs varies from platform to platform. The simplest configuration method (available on entry-level Cisco IOS switches) uses the command **port protected** entered at the interface configuration level as a way to denote isolated ports. Ports without the **port protected** command are promiscuous.

On higher-end switches, the configuration is more complex. The following Cisco CatOS example sets ports 3/2–48 as isolated ports and port 3/1 as the promiscuous port. Note the need to create two VLANs and map them together, creating the single functional PVLAN.

```
CatOS (enable) set vlan 31 pvlan primary
VTP advertisements transmitting temporarily stopped, and will resume after the
 command finishes. Vlan 31 configuration successful
CatOS (enable) show pvlan
Primary Secondary Secondary-Type  Ports
------- --------- --------------- -----------
31   -     -
CatOS (enable) set vlan 32 pvlan isolated
VTP advertisements transmitting temporarily stopped, and will resume after the
 command finishes. Vlan 32 configuration successful
CatOS (enable) set pvlan 31 32 3/2-48
Successfully set the following ports to Private Vlan 31,32:3/2-48
CatOS (enable) set pvlan mapping 31 32 3/1
Successfully set mapping between 31 and 32 on 3/1
```

There are many more options for PVLAN configuration. For more details see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_7_1/conf_gd/vlans.htm#xtocid854519.

TIP PVLANs have different functionalities depending on the switch. On some switches, PVLANs are referred to as PVLAN edge. Check the documentation for your switch to understand the specific PVLAN capabilities.

PVLAN Security Considerations

PVLANs work fine unless the attacker does some creative things with ARP to try to get past them. The basic attack is to create a static ARP entry on the compromised machine showing that the victim machine is reachable by the router's MAC address. When the frame arrives at the router, the router will notice that the packet is really destined for the victim and will happily rebuild the frame with the correct MAC address and send it on its way. This attack works only in a unidirectional fashion if the attacker has compromised only the attacking host. If both hosts are compromised, bidirectional communication is trivial to set up.

Stopping this attack is pretty easy. Configure an inbound ACL on your router to stop all traffic *from* the local subnet *to* the local subnet. For example, if your server farm segment is 172.16.34.0/24, configure the following ACL on the default gateway:

```
IOS(config)#access-list 101 deny ip 172.16.34.0 0.0.0.255
 172.16.34.0 0.0.0.255 log
IOS(config)#access-list 101 permit ip any any
IOS(config-if)#ip access-group 101 in
```

L2 Best Practices Recommendations

In summary, L2 of the OSI model can be a pretty weak link in your network security system if you aren't careful. Luckily, most of the attacks require local access, meaning the attacks are generated from the LAN they are trying to affect. Your security policy should provide guidance on how far to go in securing L2 infrastructure. Here is a summary of the best practices outlined in this section:

- Always use a dedicated VLAN ID for all trunk ports.
- Avoid using VLAN 1.
- Set all user ports to nontrunking.
- Deploy port security when possible for user ports.
- Choose one or more ARP security options.
- Enable STP attack mitigation (BPDU Guard, Root Guard).
- Use PVLANs where appropriate.
- Use MD5 authentication for VTP when VTP is needed.
- Disable CDP where it is not needed.
- Disable all unused ports and put them in an unused VLAN.
- Ensure DHCP attack prevention where needed.

IP Addressing Design Considerations

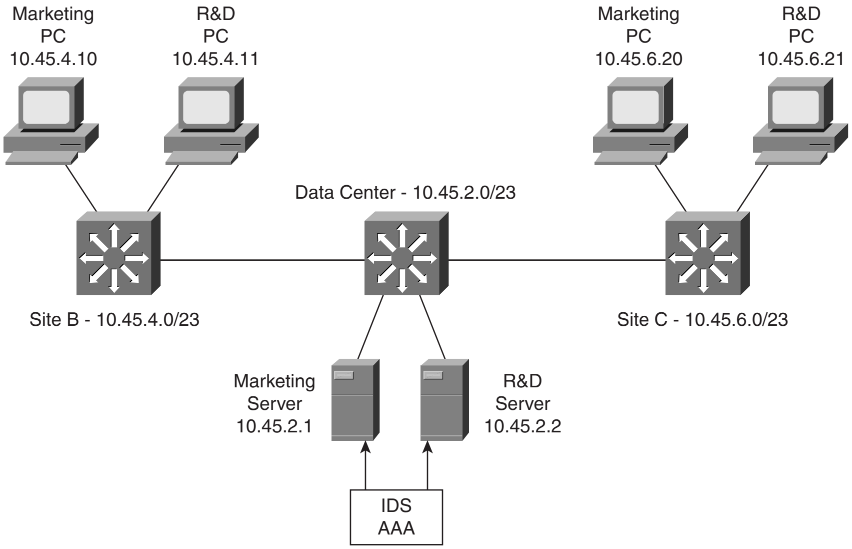
Although security considerations for L2 are important, the attacks require local access to be successful. When designing your L3 layout, the ramifications of your decisions are much more important. This section outlines overall best practices for IP addressing, including basic addressing, routing, filtering, and Network Address Translation (NAT).

General Best Practices and Route Summarization

The basic best practices for IP addressing should be familiar to you. At a high level in your design, you first must decide whether the IP address of the user on your network will have any significance from a security standpoint. For example, if you are an organization with three sites, are you just going to assign a subnet to each of the three sites, even though there are individuals at each site with different levels of security access?

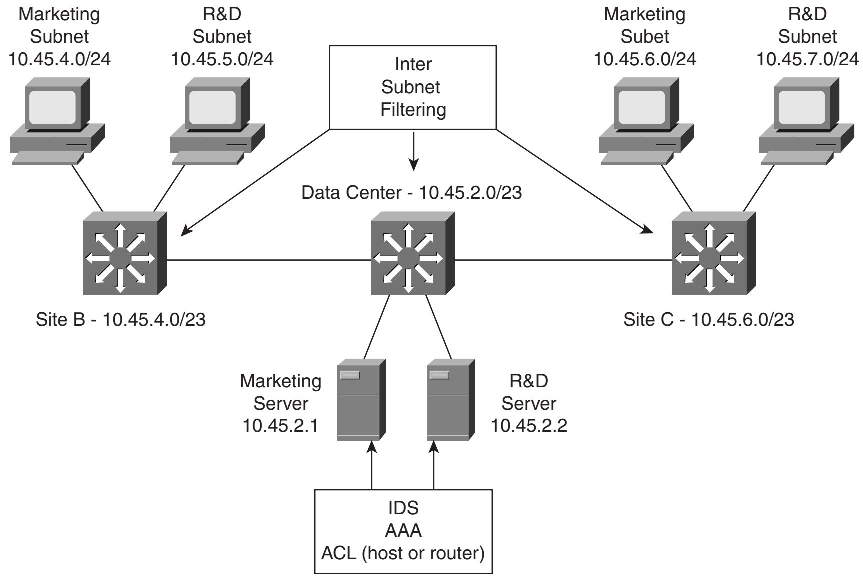
This approach is fine if your security system depends mostly on application layer security controls (AAA, intrusion detection). I've seen many designs that do this successfully, but it does take away a simple control that many find useful: L3 access control. Here, users are put into group-specific subnets that provide an additional layer of access control between the user and the resources. You can compare the two approaches in Figures 6-11 and 6-12.

Figure 6-11 *Application Security Design*



As you can see in Figure 6-11, this simplified diagram shows three sites, each with a /23 subnet of the 10 network. There are two main groups at these three sites, marketing and R&D. In this design, the servers and PCs for each of these groups share the same site-specific subnet. This means any security controls will be unable to take into account the IP address of the system attempting access.

Figure 6-12 *Application Security Plus L3 ACL Design*



In Figure 6-12, the same /23 subnet exists per site, but it has been further divided into two /24 subnets, one for each organization. This allows intersubnet filtering at the routed connection at each site. This filtering could be used in sites B and C to prevent the R&D and marketing departments from accessing each other's PCs and at the data center to ensure that only marketing PCs can access marketing servers and likewise for the R&D department.

TIP This sort of filtering is referred to as role-based subnetting throughout the rest of this book.

Although the benefits of the approach shown in Figure 6-12 are pretty clear, this kind of design gets exponentially more difficult as the number of sites of different groups in an organization increase. Wireless features and the wired mobility of the workforce also affect the feasibility of this design. Technologies such as 802.1x (see Chapter 9) can make this easier but by no means solve all the problems.

Like any of the discussions in this book, every design decision should come back to the requirements of your security policy.

TIP I have seen some designs that attempt to trunk VLANs throughout the sites of an organization. For example, consider if the design in Figure 6-12 had only two subnets, one for R&D and another for marketing. These subnets would need to exist at all three sites. The principal problem with this design is the need to trunk the VLANs at L2 throughout the organization. This increases the dependence on STP and could make the design more difficult to troubleshoot.

Route summarization is always something that sounds easy when you first design a network and gets harder and harder as the network goes into operation. The basic idea is to keep your subnet allocations contiguous per site so that your core routers need the smallest amount of routes in their tables to properly forward traffic. In addition to reducing the number of routes on your routers, route summarization also makes a network far easier to troubleshoot. In Figure 6-12, you can see a very simple example of route summarization. Sites B and C each have two /24 subnets, but they are contiguously addressed so they can be represented as one /23 subnet. These summarizations also help when writing ACLs since a large number of subnets can be identified with a single summarized ACL entry.

Ingress/Egress Filtering

Ingress/egress filtering is different from what you would normally call firewalling. Ingress/egress filtering is the process of filtering large classes of networks that have no business being seen at different parts of your network. Although ingress/egress can mean different things depending on your location, in this book *ingress* refers to traffic coming into your organization, and *egress* refers to traffic leaving it. Several types of traffic can be filtered in this way, including RFC 1918 addresses, RFC 2827 antispoof filtering, and nonroutable addresses. The next several sections discuss each option as well as a method of easily implementing filtering using a feature called verify unicast reverse path forwarding (uRPF).

RFC 1918

RFC 1918, which can be downloaded from http://www.ietf.org/rfc/rfc1918.txt, states that a block of addresses has been permanently set aside for use in private intranets. Many organizations today use RFC 1918 addressing inside their organizations and then use NAT to reach the public Internet. The addresses RFC 1918 sets aside are these:

10.0.0.0–10.255.255.255 (10/8 prefix)
172.16.0.0–172.31.255.255 (172.16/12 prefix)
192.168.0.0–192.168.255.255 (192.168/16 prefix)

The basic idea of RFC 1918 *filtering* is that there is no reason you should see RFC 1918 addressing from outside your network coming in. So, in a basic Internet design, you should block RFC 1918 addressing before it crosses your firewall or WAN router. An ACL on a Cisco router to block this traffic looks like this:

```
IOS(config)#access-list 101 deny ip 10.0.0.0 0.255.255.255 any log
IOS(config)#access-list 101 deny ip 172.16.0.0 0.15.255.255 any log
IOS(config)#access-list 101 deny ip 192.168.0.0 0.0.255.255 any log
IOS(config)#access-list 101 permit ip any any
IOS(config-if)#ip access-group 101 in
```

This ACL stops any traffic with a source IP address in the RFC 1918 range from entering your site. Also, your Internet service provider (ISP) should be blocking RFC 1918 addressing as well; check to make sure it is.

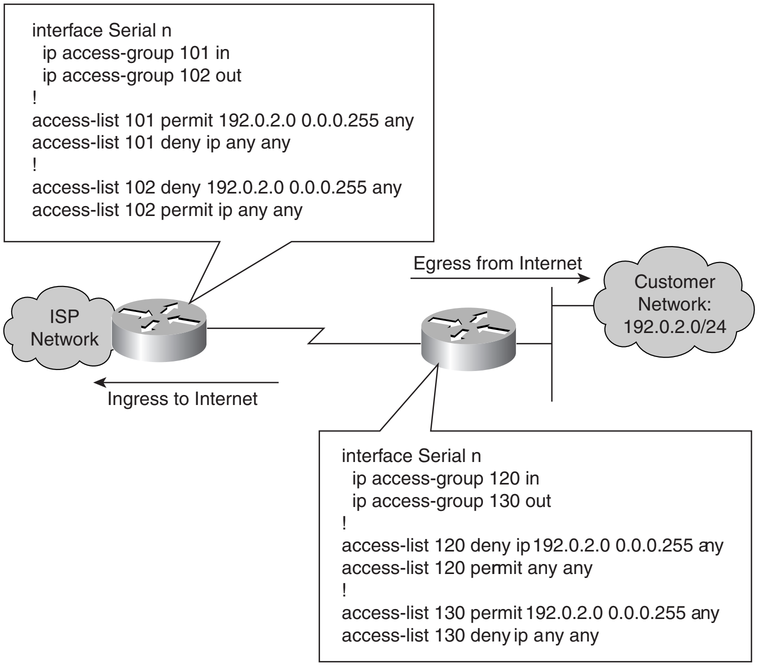
I had a conversation once with the administrator of a popular website who was the victim of a distributed denial of service (DDoS) attack that was launched entirely from RFC 1918 address space. If only his ISP had blocked this space, his website would have been unaffected. You can bet he had some choice words for the ISP after this attack!

One consideration with RFC 1918 addressing is the headaches it can cause when you need to connect to another organization that uses the same range of RFC 1918 addresses. This can happen through a merger or in an extranet arrangement. To at least slightly reduce the chances of this, pick addresses that aren't at the beginning of each major net range. For example, use 10.96.0.0/16, not 10.1.0.0/16.

RFC 2827

RFC 2827 defines a method of ingress and egress filtering based on the network that has been assigned to your organization. If your organization is assigned the 192.0.2.0/24 address, those are the only IP addresses that should be used in your network. RFC 2827 filtering can ensure that any packet that leaves your network has a source IP address of 192.0.2.0/24. It can also make sure that any packet entering your network has a source IP address *other than* 192.0.2.0/24. Figure 6-13 shows how this filtering could be applied both at the customer network and the ISP.

Figure 6-13 *RFC 2827 Filtering*



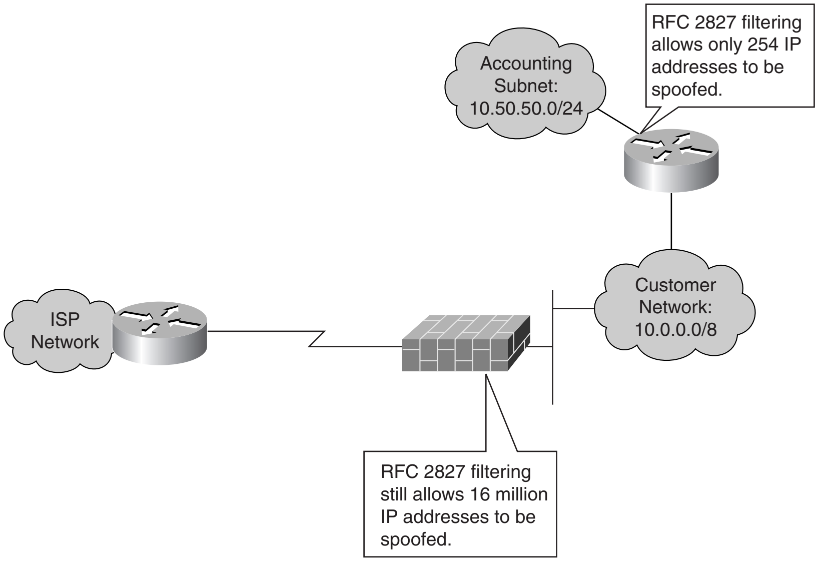
```
interface Serial n
  ip access-group 101 in
  ip access-group 102 out
!
access-list 101 permit 192.0.2.0 0.0.0.255 any
access-list 101 deny ip any any
!
access-list 102 deny 192.0.2.0 0.0.0.255 any
access-list 102 permit ip any any
```

```
interface Serial n
  ip access-group 120 in
  ip access-group 130 out
!
access-list 120 deny ip 192.0.2.0 0.0.0.255 any
access-list 120 permit any any
!
access-list 130 permit 192.0.2.0 0.0.0.255 any
access-list 130 deny ip any any
```

When implementing RFC 2827 filtering in your own network, it is important to push this filtering as close to the edge of your network as possible. Filtering at the firewall only might allow too many different spoofed addresses (thus complicating your own trace back). Figure 6-14 shows filtering options at different points in a network.

WARNING Be careful about the potential performance implications of RFC 2827 filtering. Make sure the devices you are using support hardware ACLs if your performance requirements dictate that they must. Even with hardware ACLs, logging is generally handled by the CPU, which can adversely affect performance when you are under attack.

Figure 6-14 *Distributed RFC 2827 Filtering*


Accounting Subnet: 10.50.50.0/24
RFC 2827 filtering allows only 254 IP addresses to be spoofed.
ISP Network
Customer Network: 10.0.0.0/8
RFC 2827 filtering still allows 16 million IP addresses to be spoofed.

When using RFC 2827 filtering near your user systems and those systems that use DHCP, you must permit additional IP addresses in your filtering. Here are the details, straight from the source (RFC 2827):

> If ingress filtering is used in an environment where DHCP or BOOTP is used, the network administrator would be well advised to ensure that packets with a source address of 0.0.0.0 and a destination of 255.255.255.255 are allowed to reach the relay agent in routers when appropriate.

If properly implemented, RFC 2827 can reduce certain types of IP spoofing attacks against your network and can also prevent IP spoofing attacks (beyond the local range) from being launched against others from your site. If everyone worldwide implemented RFC 2827 filtering, the Internet would be a much safer place because hiding behind IP spoofing attacks would be nearly impossible for attackers.

Nonroutable Networks

Besides private network addressing and antispoof filtering, there are a host of other networks that have no business being seen, including those that won't be seen for some time because they haven't yet been allocated. For example, at the time this was written, the /8 networks from 82 to 126 had not yet been allocated from the Internet Assigned Numbers Authority (IANA) to any of the regional Internet registries (RIRs). This data can be tracked at a very high level at the following URL: http://www.iana.org/assignments/ipv4-address-space.

IANA is responsible for allocating Internet Protocol version 4 (IPv4) address space to the RIRs; the RIRs then allocate address space to customers and ISPs. All of this is of interest to ISPs, which might try not to forward traffic from address space that has yet to be allocated. In doing so, they will reduce the available networks that attackers can use in spoofing attacks. Rob Thomas (founder of Cymru.com) maintains an unofficial page of something called "bogon" ranges. Bogon ranges are address ranges that have no business being seen on the Internet. They are either reserved, specified for some special use, or unallocated to any RIR. Filtering this comprehensive list of subnets can narrow the potential source of spoofed IP packets. Be aware that this list can change every few months, meaning these filters must also be periodically changed. Thomas's bogon list is available here: http://www.cymru.com/Documents/bogon-list.html.

WARNING Be aware that, with all this filtering, you are making things more difficult for the attacker but not impossible. An attack tool could easily decide to spoof only address ranges that have been allocated. In fact, attackers have started to reduce the amount of attacks that they actually spoof traffic from. By compromising several other hosts around the world, they feel safe launching attacks from those remote systems directly.

For organizations unable to track the changing bogon list, RFC 3330 states some special subnets that can permanently be filtered from your network. They are the following:

- **0.0.0.0/8**—This network refers to hosts on this network, meaning the network where the packet is seen. This range should be used only as a source address and should not be allowed in most situations except DHCP/Bootstrap Protocol (BOOTP) broadcast requests and other similar operations. It can certainly be filtered inbound and outbound from your Internet connection.
- **127.0.0.0/8**—This subnet is home to the address 127.0.0.1, referring to localhost, or your machine. Packets should never be seen on networks sourced from the 127/8 network. RFC 3330 says it best: "No addresses within this block should ever appear on any network anywhere."
- **169.254.0.0/16**—This subnet is reserved for hosts without access to DHCP to autoconfigure themselves to allow communications on a local link. You are safe in filtering this subnet on your network.
- **192.0.2.0/24**—This subnet, commonly called the "TEST-NET," is a /24 subnet allocated for sample code and documentation. You might notice that some diagrams in this book use this address when a registered address is represented. You should never see this network in use anywhere.

- **198.18.0.0/15**—This subnet has been set aside for performance benchmark testing as defined in RFC 2544. Legitimate network-attached devices should not use this address range, so it can be filtered.
- **224.0.0.0/4**—This is the multicast range. On most networks, this can be filtered at your Internet edge, but obviously, if your network supports multicast, this would be a bad idea. In most cases, it is a bad idea to filter it internally because multicast addresses are used for many popular routing protocols.
- **240.0.0.0/4**—This is the old Class E address space. Until IANA decides what to do with this space, it can be safely filtered at your Internet edge.

TIP One thing to consider is that although filtering bogons or the subset listed in RFC 3330 is possible on your internal network, it isn't necessary. If you are properly implementing RFC 2827 filtering, you implicitly deny any network that is not your own, which would include all bogon ranges. Filtering bogons is most appropriate at your Internet edge, where you would also implement RFC 2827 and RFC 1918 filtering. Within your campus network, filtering with RFC 2827 is sufficient.

uRPF

A far easier way to implement RFC 2827 filtering is to use something that on Cisco devices is called verify unicast reverse path forwarding (uRPF). This functionality is available on multiple vendors' platforms, though it might be known by a different name. Cisco documentation for basic uRPF configuration can be found here: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fothersf/scfrpf.htm.

This filtering works by blocking inbound source IP addresses that don't have a route in the routing table pointing back at the same interface on which the packet arrived. To be more specific, uRPF checks the forwarding information base (FIB) that is created from the routing table on all Cisco devices running Cisco express forwarding (CEF). As such, uRPF works only on Cisco devices that support CEF. For example, consider the following situation:

1. A packet arrives on a router with a source IP address of 192.0.2.5 at interface Ethernet0/0.
2. uRPF checks the FIB by doing a reverse lookup to determine whether the return path to the source IP address would use the same interface that the packet arrived on. If the best return path would use a different interface, the packet is dropped.
3. If the best return path is the interface that the packet arrived on, the packet is forwarded toward its destination.

If there are multiple best paths (as in the case of load balancing), uRPF will forward the packet as long as the interface the packet arrived on is in the list of best paths.

The syntax for uRPF is very simple:

```
IOS(config-if)#ip verify unicast reverse-path
```

Some additional options are available to provide more granularity in configuration. For example, ACLs that are evaluated when a uRPF check fails can be applied.

WARNING Like RFC 2827, uRPF is most effective close to the edge of your network. This is because the edge of your network is the most likely location in your network to have routing symmetry (meaning packets arrive on the same interface that the return traffic will use). You should not deploy uRPF on interfaces that contain asymmetrically routed traffic, or legitimate traffic will be dropped.

For a service provider or a very large enterprise customer, there is an additional option called uRPF loose mode. (The previous mode is sometimes called strict mode.) Loose mode allows a packet to forward as long as there is a return route to the source *somewhere* in the FIB. This has the result of blocking the entire bogon list. I say "larger enterprise" here because you generally need the entire Border Gateway Protocol (BGP) routing table on your router before this is useful; otherwise, any spoofed packet will have an entry for the FIB because of the default route. When you have the entire BGP routing table on a device, you usually don't need a default route. The command for loose-mode uRPF looks like this:

```
IOS(config-if)# ip verify unicast source reachable-via any
```

There is also an *allow-default* flag that can be set, depending on whether you want the default route to be considered a valid route when making the uRPF decision.

NAT

Few technologies have generated as much discussion among security communities as NAT. The idea of translating private addresses to public addresses is seen by many as a good way for organizations without their own IP ranges to get on the Internet. Rather than address their internal network with the addresses provided by their ISP (thus making changing ISPs very difficult), they simply choose to translate RFC 1918 addresses as they leave the network. If you want a simple rule to follow, never use NAT in a security role. NAT is fine for its intended purpose: address translation. But if you have places in your network where your security relies on NAT, you probably need to reevaluate your design. If you agree with me and understand why, feel free to skip the rest of this section; if not, read on.

NAT can be done in three main ways: *static translation* is when an internal IP address corresponds to a specific external IP address. This is generally done for publicly accessible servers that must be reachable from the outside on a predictable IP address. *One-to-one NAT,* or *basic NAT,* is when an IP address inside corresponds to a single address on the outside selected from a pool. One day a system might get 192.0.2.10; another day it might get .11. Finally, *many-to-one NAT* (sometimes called port address translation [PAT] or NAT overload) is when a large number of private addresses can be translated to a single public IP address. This is a very popular use of NAT for organizations with limited public address space. All of their internal users can use a small number of public IP addresses.

There is little debate about the security benefits of the first two NAT options (static and one-to-one): simply put, there is none. Because each internal address corresponds to a single public address, an attacker would merely need to attack the public address to have that attack translated to the private address. Where the discussion comes in is about the benefits of many-to-one NAT. Although their numbers seem to be declining, there are some who believe that many-to-one NAT is a valuable security tool. The basic premise is this: when you are using many-to-one NAT, the NAT system keeps track of user connections from the inside to the outside by changing the source port number at Layer 4 (L4). When the return traffic comes back destined for that port number/IP address (with the right source IP address and port number), the NAT system translates the destination port number/IP address to the internal private host and sends the traffic.

This type of protection falls into the security-through-obscurity category outlined in Chapter 1, "Network Security Axioms." An attacker could do a number of things that would not be stopped by NAT:

- Send a Trojan application by e-mail (or a compromised web page) that opens a connection from the private host to the attacking host.
- Send data with the correct port number and IP address (with the right spoofed source port and IP address). This would require some trial and error on the part of the attacker, but it cannot be discounted in the event that the attacker is going after your network specifically.
- Allow outside connections. Although this isn't so much an attacker action, there are many applications on a host that open periodic connections with hosts on the Internet. NAT has no way of blocking connections from the inside out.

The specifics of when, where, and why to use NAT are general networking design issues, not security related. From a security perspective, I would have no reservations using public addresses for all of a network and not using NAT at all.

TIP For teleworkers and home users, sometimes NAT might be the only network-level security technology available. In these cases, many-to-one NAT is certainly better than no network security technology. However, it is better to properly secure the hosts on a network and have no NAT than to have NAT without any host security protections.

ICMP Design Considerations

One way to spot inexperienced secure network design is to look for networks that completely block Internet Control Message Protocol (ICMP) traffic. As any operator of all but the smallest networks will tell you, troubleshooting a network without ping is very frustrating, bordering on impossible. That said, ICMP messages should not be enabled everywhere without reservation. Some security considerations must be understood, just like for any other protocol. This section assumes basic ICMP understanding. Refer to your favorite TCP/IP book for background or read RFC 792.

ICMP security can be a very lengthy discussion because lots of nasty things can be done with ICMP messages when scanning networks or trying to gain a covert channel. If you are interested in this sort of thing, Ofir Arkin's paper titled "ICMP Usage in Scanning" is available at http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.5.pdf. Rob Thomas has some guidelines for ICMP filtering that are available here: http://www.cymru.com/Documents/icmp-messages.html.

The basics behind ICMP design considerations are to define how much ICMP traffic you should allow on your network and which messages types you should filter.

ICMP Rate Limiting

Because ICMP is a troubleshooting and error-reporting tool, there should be a limit to the amount of ICMP traffic you see on a given network. For example, on a 100 Mbps Ethernet link, you might block ICMP traffic that exceeds 500 Kbps. A technology called committed access rate (CAR) enables this sort of filtering and is discussed later in this chapter.

ICMP Message Type Filtering

As Chapter 2 discussed, your own security policies and threat models might be different from those assumed here. Deploying filters throughout your internal network to permit only the ICMP message types required would be difficult. As a first step, focus on possible boundaries of trust between two networks. Your network will have its own trust boundaries, but here are a few to get you started. Zones of trust are detailed more fully in Chapter 12, "Designing Your Security System."

- Internet and internal network
- Management network and production network
- Critical applications and production network

An easy first step in ICMP filtering is to deny any ICMP message that is a fragment. First, the ICMP messages you must permit are generally small. Echo and echo reply, for example, default on BSD UNIX to 84 bytes: 20-byte IP header, 8-byte ICMP header, and 56 bytes of ICMP data.

Other required ICMP messages are similarly small and come nowhere near the minimum link size on today's IP networks. Blocking ICMP fragments is easy using an ACL:

```
access-list 101 deny icmp any any fragments
```

WARNING The *fragments* keyword in a Cisco ACL has some special use rules. For a detailed discussion of this, including flow charts and examples, check the paper at the following URL: http://www.cisco.com/warp/public/105/acl_wp.html.

As a quick summary of the paper, the *fragments* keyword applies only to noninitial fragments (fragment offset > 0), so in the preceding example, the first part of a fragmented ICMP packet will not match that entry, while all subsequent fragments will.

When filtering ICMP messages between trust boundaries, apply the security principle "Expressly permit, implicitly deny." Though your specific requirements may vary, the following ICMP types should be permitted in some form:

- ICMP echo request and ICMP echo reply
- ICMP destination unreachable—fragmentation needed but DF bit set
- ICMP time exceeded

ICMP Echo Request and ICMP Echo Reply

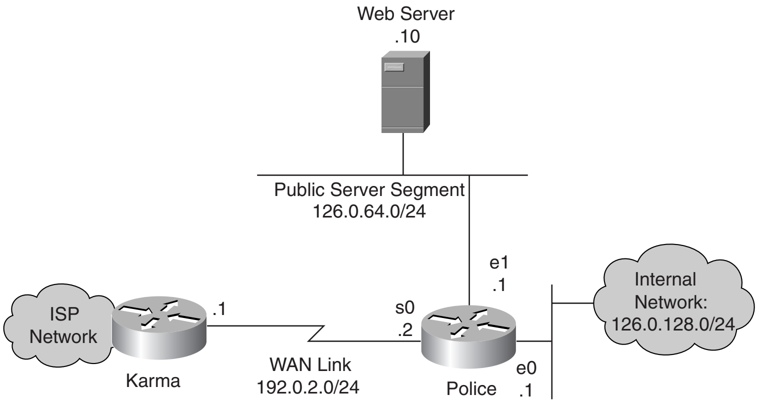
ICMP echo request (Type 8 Code 0) and ICMP echo reply (Type 0 Code 0) are better known as the message types used by the **ping** command. The format of an ICMP echo message has the standard 8 bytes of ICMP header information and then allows for a variable-length data field that can contain any kind of data. Certain size ping packets caused system crashes on some older OSs. This attack was commonly called the Ping of Death. More information can be found here: http://www.insecure.org/sploits/ping-o-death.html. Permitting ICMP echo can lead to DoS attacks and buffer overflows as discussed in Chapter 3. It can also lead to a covert channel because information can be embedded into the data field in the ICMP echo message. An attacker that installs special software on a host internal to your network could communicate back and forth using only ICMP echo request or reply messages. Covert channels have been implemented in many different protocols, and they are impossible to completely eliminate. So, with these risks, it is understandable why a security engineer would want to stop ICMP echo messages. Unfortunately, troubleshooting would be far too difficult without it making your overall network less secure in most cases. With all that said, here are the best practices:

- Permit ICMP echo request messages to leave your network destined for any network you have reason to communicate with.

- Permit ICMP echo reply messages to your internal hosts from any network you have reason to communicate with.
- Permit ICMP echo request messages from external hosts to servers they must access (public web servers, for example). As of this writing, a random sampling of top websites yielded several that block inbound pings to their servers and several more that permit them. As an organization, you must weigh the risks of allowing this traffic against the risks of denying this traffic and causing potential users troubleshooting difficulties.
- Permit ICMP echo reply messages from any server system to the networks where that server's users reside. Echo replies from your public web server to the Internet at large is an example of this.
- Deny every other ICMP echo message.

As an example, consider the very simplified Internet edge shown in Figure 6-15.

Figure 6-15 *Simple Internet Edge*



If you were writing ICMP echo access lists for router "police," the inbound Serial0 ACL would look like this:

```
! permit echo-request to Serial0 interface of the router
access-list 101 permit icmp any host 192.0.2.2 echo
! permit echo-request to public server
access-list 101 permit icmp any host 126.0.64.10 echo
! permit echo-reply from anywhere to the internal network and the public server
access-list 101 permit icmp any 126.0.128.0 0.0.0.255 echo-reply
access-list 101 permit icmp any host 126.0.64.10 echo-reply
```

The ACL on the inbound Ethernet0 interface would look like this:

```
! permit echo-request from the internal network to anywhere
access-list 102 permit icmp 126.0.128.0 0.0.0.255 any echo
```

The ACL on the inbound Ethernet1 interface would look like this:

```
! permit echo-request from the public web server to anywhere
access-list 103 permit icmp host 126.0.64.10 any echo
! permit echo-reply from the public web server to anywhere
access-list 103 permit icmp host 126.0.64.10 any echo-reply
```

Based on these ACLs, internal users can ping the web server and the Internet, the Internet can ping the web server, and the web server can ping the Internet. Of special note is that the web server cannot ping internal hosts. Based on your security policies, you can permit this to aid in troubleshooting, but be aware that many organizations consider public servers to be not much more trusted than the Internet. To make the change, you would add this line to the Ethernet0 ACL:

```
access-list 102 permit icmp 192.0.128.0 0.0.0.255 host 192.0.64.10 echo-reply
```

NOTE Cisco router ACLs can be applied inbound or outbound on a given interface. Security folks, myself included, tend to prefer inbound ACLs, but there are situations in which you must use both and situations in which an outbound ACL makes more sense. I prefer inbound because the packets are blocked before they cross the router. Outbound ACLs allow the packet to be routed by the router and then are blocked when they try to leave. This could leave the router open to certain attacks.

Another special note on Cisco ACLs is that ACLs never apply to traffic generated by the router. So, even if you have an inbound and an outbound ACL on a router denying all traffic, the router will still be able to send any packet it wants; the return packet, however, will be blocked as usual.

ICMP Destination Unreachable—Fragmentation Needed but DF Bit Set

ICMP destination unreachable messages (type 3 code 0–15) are a whole range of messages designed to alert the sending system that something is wrong with a particular message sent. This includes specific errors such as network unreachable (code 0), host unreachable (code 1), protocol unreachable (code 2), and port unreachable (code 3). These types of messages are generated by hosts and routers when a sending system tries to go somewhere that is unreachable for whatever reason. Many security administrators block most type 3 messages because the sending host will often figure out that the service is unavailable on its own without the benefit of the ICMP message (albeit more slowly). One message is required though: “fragmentation needed but DF bit set” (type 3 code 4). This message is required for path Maximum Transmission Unit (MTU) discovery to work. Path MTU discovery is the method most hosts use to determine the IP MTU size for their traffic. Without it functioning properly, large TCP segments could be dropped without a means to remedy the problem because the offending host never knows why the drop occurs.

Path MTU discovery has some interesting implications in IPsec and is discussed in more detail in Chapter 10, “IPsec VPN Design Considerations.”

ICMP type 3 code 4 messages can be easily permitted by adding the following line to the ACLs built for Figure 6-15:

```
access-list 101 permit icmp any any packet-too-big
```

ICMP Time Exceeded

ICMP time exceeded: Time-to-Live (TTL) equals 0 during transit (type 11 code 0) is required because it is used by traceroute. To permit these messages, add the following line to the ICMP ACLs you have seen in this section:

```
access-list 101 permit icmp any any time-exceeded
```

ICMP Filtering Recommendations

As you can see, there was a reason that ICMP was created beyond as a playground for attackers. Although most of the 15 ICMP message types can be blocked, several are necessary to the healthy operation of a network. We can rebuild the previous ACLs to allow all the messages we discussed, to block fragments, and to deny any other ICMP messages. Those ACLs are as follows.

Router “police” Serial0 ACL, inbound:

```
! deny non-initial ICMP Fragments
access-list 101 deny icmp any any fragments
! permit echo-request to Serial0 interface of the router
access-list 101 permit icmp any host 192.0.2.2 echo
! permit echo-request to public server
access-list 101 permit icmp any host 126.0.64.10 echo
! permit echo-reply from anywhere to the internal network and the public server
access-list 101 permit icmp any 126.0.128.0 0.0.0.255 echo-reply
access-list 101 permit icmp any host 126.0.64.10 echo-reply
! permit "fragmentation needed but DF bit set" message
access-list 101 permit icmp any any packet-too-big
! permit "Time exceeded" message
access-list 101 permit icmp any any time-exceeded
! deny any other ICMP message
access-list 101 deny icmp any any
! from here you would continue with other non ICMP related ACL entries
```

Router “police” Ethernet0 ACL, inbound:

```
! deny non-initial ICMP Fragments
access-list 102 deny icmp any any fragments
! permit echo-request from the internal network to anywhere
access-list 102 permit icmp 126.0.128.0 0.0.0.255 any echo
! permit "fragmentation needed but DF bit set" message
access-list 102 permit icmp any any packet-too-big
```

```
! permit "Time exceeded" message
access-list 102 permit icmp any any time-exceeded
! deny any other ICMP message
access-list 102 deny icmp any any
! from here you would continue with other non ICMP related ACL entries
```

Router "police" Ethernet1 ACL, inbound:

```
! deny non-initial ICMP Fragments
access-list 103 deny icmp any any fragments
! permit echo-request from the public web server to anywhere
access-list 103 permit icmp host 126.0.64.10 any echo
! permit echo-reply from the public web server to anywhere
access-list 103 permit icmp host 126.0.64.10 any echo-reply
! permit "fragmentation needed but DF bit set" message
access-list 103 permit icmp any any packet-too-big
! permit "Time exceeded" message
access-list 103 permit icmp any any time-exceeded
! deny any other ICMP message
access-list 103 deny icmp any any
! from here you would continue with other non ICMP related ACL entries
```

NOTE If you want to get very picky, you could probably block the packet-too-big and time-exceeded messages from being generated by either the public server segment or the internal network, depending on the rest of your configuration. With protocols such as ICMP (which are often used in troubleshooting), you are probably better off following the KISS principle by making your ICMP filtering consistent as much as possible.

Routing Considerations

As we continue to slowly work our way up the OSI model with these best practices, it is now useful to develop some design considerations in the realm of routing. The most important is the security of the routing protocol.

Routing Protocol Security

Routing security has received varying levels of attention over the past several years and has recently begun to attract more attention specifically around BGP on the public Internet. Despite this new attention, however, the area most open to attack is often not the Internet's BGP tables but the routing systems within your own enterprise network. Because of some of the sniffing-based attacks discussed in Chapter 3 and earlier in this chapter, an enterprise routing infrastructure can easily be attacked with MITM and other attacks designed to corrupt or change the routing tables with the following results:

- **Traffic redirection**—In this attack, the adversary is able to redirect traffic, enabling the attacker to modify traffic in transit or simply sniff packets.

- **Traffic sent to a routing *black hole***—Here the attacker is able to send specific routes to null0, effectively kicking IP addresses off of the network.
- **Router DoS**—Attacking the routing process can result in a crash of the router or a severe degradation of service.
- **Routing protocol DoS**—Similar to the attack previously described against a whole router, a routing protocol attack could be launched to stop the routing process from functioning properly.
- **Unauthorized route prefix origination**—This attack aims to introduce a new prefix into the route table that shouldn't be there. The attacker might do this to get a covert attack network to be routable throughout the victim network.

There are four primary attack methods for these attacks:

- Configuration modification of existing routers
- Introduction of a rogue router that participates in routing with legitimate routers
- Spoofing a valid routing protocol message or modifying a valid message in transit
- Sending of malformed or excess packets to a routing protocol process

These four attack methods can be mitigated in the following ways:

- To counter configuration modification of existing routers, you must secure the routers. This includes not only the configuration of the router but also the supporting systems it makes use of, such as TFTP servers. See Chapter 5, "Device Hardening," for more information.
- Anyone can attempt to introduce a rogue router, but to cause damage, the attacker needs the other routing devices to believe the information that is sent. This can most easily be blocked by adding message authentication to your routing protocol. More on this subject can be found in the next section. Additionally, the routing protocol message types can be blocked by ACLs from networks with no need to originate them.
- Message authentication can also help prevent the spoofing or modification of a valid routing protocol message. In addition, the transport layer protocol (such as TCP for BGP) can further complicate message spoofing because of the difficulty in guessing pseudorandom initial sequence numbers (assuming a remote attacker).
- Excess packets can be stopped through the use of traditional DoS mitigation techniques, which are discussed later in the chapter. Malformed packets, however, are nearly impossible to stop without the participation of the router vendor. Only through exhaustive testing and years of field use do routing protocol implementations correctly deal with most malformed messages. This is an area of computer security that needs increased attention, not just in routing protocols but in all network applications.

As you can see, stopping all these attacks is not a matter of flipping on the secure option in your routing protocols. As stated in Chapter 2, you must decide for your own network what threats need to be stopped. In addition to the specific threats mentioned here, it is also very useful to follow the network design best practices of not running routing protocols on interfaces with no reason to route and of using distribution lists to limit the routing prefixes that are sent or received by a specific routing instance. Details on distribution lists can be found in your favorite Internet routing book.

Routing Protocol Message Authentication

Although they vary in the strength of the authentication they offer, nearly all routing protocols support some form of message authentication. There are two principal types of authentication used in routing protocols today: plaintext password and MD5 digest.

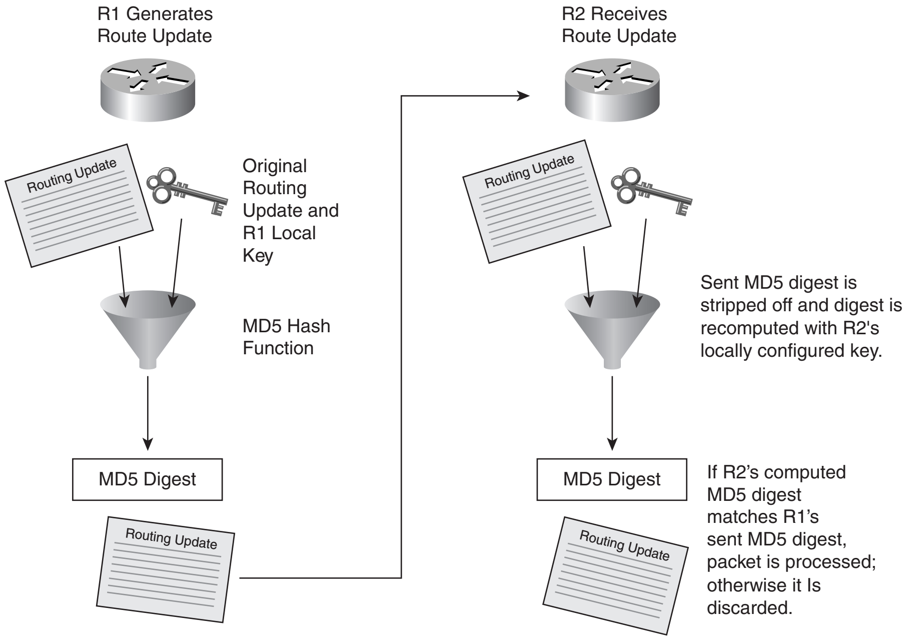
Plaintext Password Authentication

Plaintext password authentication is just what it sounds like. A password is attached to the routing update and is sent in the clear along with the routing update. The passwords have specific length requirements as defined by the routing protocol in use. Plaintext password authentication should be considered specious security because anyone who sees a single routing update on the wire sees the authentication information if it is in use. From this point on, the attacker can appear to be a member of the trusted routing domain. The plaintext password does offer some benefit in that it prevents routing protocol changes when an invalid router is *accidentally* introduced into a production routing environment.

MD5 Digest Authentication

MD5 digest works by creating a 16-byte hash of the routing message combined with a secret key. The 16-byte value is, therefore, message-specific, and modification of the message by an attacker invalidates the 16-byte digest appended to the message. Without the secret key, which is never sent over the wire by the routing protocol, the attacker is unable to reconstruct a valid message. It is worth noting that the MD5 option provides authentication and packet integrity, not confidentiality. Figure 6-16 shows how the hash function operates.

Figure 6-16 *MD5 Digest for Routing Authentication*



WARNING MD5 passwords should have the same properties as other critical passwords in your network. They should follow the password creation guidelines in your security policy. If you choose a weak password, it is possible for an attacker to use brute-force guessing to determine your digest password, thereby allowing the attacker to become a trusted member of the routing domain.

Specific Routing Protocol Security Options

This section details the security options available in the most widely used routing protocols.

Routing Information Protocol

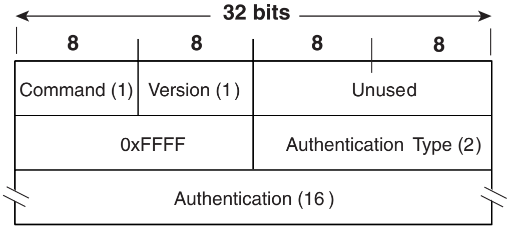
Routing Information Protocol (RIP) version 1 (RFC 1058) has no mechanism whatsoever to authenticate routing messages. As such, it should never be used in security-sensitive environments.

RIP v2

RIP v2 (RFC 1723) supports a 16-byte plaintext password that can be attached to routing updates. RFC 2082 specifies a proposed standard for adding MD5 authentication to RIP v2. Whenever possible, use the MD5 digest instead of the basic password.

RIP v2 plaintext messages have the format shown in Figure 6-17.

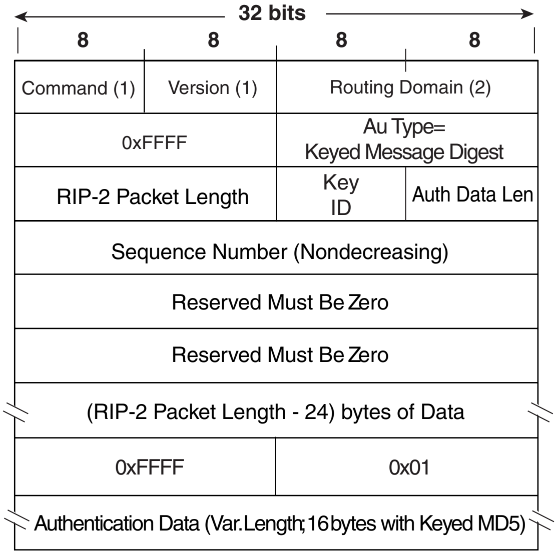
Figure 6-17 *RIP v2 Plaintext Authentication*



8	8	8	8
Command (1)	Version (1)	Unused	
0xFFFF		Authentication Type (2)	
Authentication (16)			

RIP v2 MD5 authenticated messages have the format shown in Figure 6-18.

Figure 6-18 *RIP v2 MD5 Authentication*



8	8	8	8
Command (1)	Version (1)	Routing Domain (2)	
0xFFFF		Au Type= Keyed Message Digest	
RIP-2 Packet Length		Key ID	Auth Data Len
Sequence Number (Nondecreasing)			
Reserved Must Be Zero			
Reserved Must Be Zero			
(RIP-2 Packet Length - 24) bytes of Data			
0xFFFF		0x01	
Authentication Data (Var.Length; 16 bytes with Keyed MD5)			

The configuration for RIPv2 authentication is as follows:

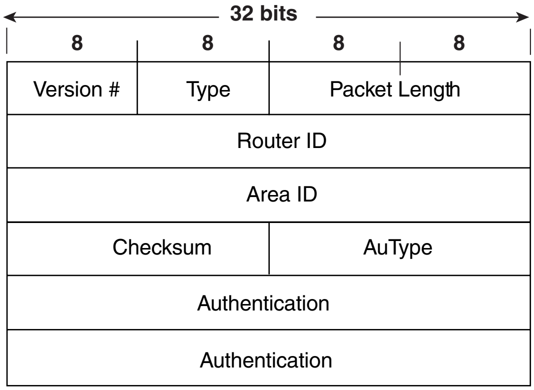
```
!Enable RIP authentication
Router(config-if)# ip rip authentication key-chain name-of-chain
!Specify authentication type
Router(config-if)# ip rip authentication mode {text ¦ md5}
!Identify key chain
Router(config)# key chain name-of-chain
!Specify key number
Router(config-keychain)# key number
!Specify actual key
Router(config-keychain-key)# key-string text
```

Open Shortest Path First

Open Shortest Path First (OSPF) (RFC 2328) is one of the most widely used interior gateway protocols today. It supports nearly every bell and whistle you could ask of your routing protocol. On the security side, it offers both plaintext authentication (with basic message checksum) and the much more secure MD5 digest.

OSPF MD5 authenticated messages have the format shown in Figure 6-19.

Figure 6-19 *OSPF Packet Header*



Note that there is no special format for OSPF when you use authentication. Authentication is assumed, even though it defaults to null authentication. In Figure 6-19, *AuType* specifies the authentication type.

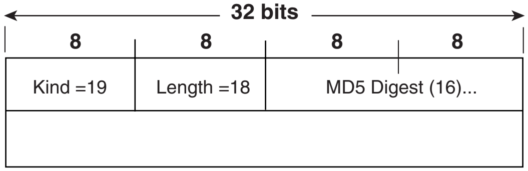
The configuration for OSPF MD5 authentication is as follows:

```
!The MD5 key is always defined per interface but enabling MD5 can be
!done either on the interface as shown in the first command
!or at the area as in the second command. The
!third command is required for both options.
!Specify OSPF authentication type
Router(config-if)# ip ospf authentication message-digest
!Enable MD5 for an area
Router(config-router)# area area-id authentication message-digest
!Specify MD5 key
Router(config-if)# ip ospf message-digest-key key-id md5 key
```

BGP

BGP is most widely used in routing between two different routed domains, such as between you and your ISP or your ISP and the upstream ISP. BGP supports MD5 authentication. Note that because BGP uses TCP as a transport protocol, the MD5 authentication is done as a TCP option. More details on this can be found in RFC 2385. TCP Option 19 is specified for this authentication and takes the format specified in Figure 6-20.

Figure 6-20 *TCP Option 19 for BGP MD5 Authentication*



The configuration for BGP MD5 authentication is as follows:

```
! Enable TCP MD5 authentication for a specific neighbor
Router(config-router)# neighbor neighbor_ip_addr password text
```

NOTE As of this writing, BGP security is receiving a fair amount of attention in the industry. Several extensions are being proposed to allow the BGP messages to be authenticated, as well as to check that an advertiser of a particular prefix is authorized to do so. Most of these mechanisms make at least partial use of a Public Key Infrastructure (PKI). These options will take some time to be agreed upon; in the meantime, best practices are the best line of defense.

Because BGP is unicast as opposed to broadcast or multicast, IPsec can be used with it to provide even greater security. As of this writing, some networks were in the testing phase of their deployments. I would recommend waiting until IPsec in combination with BGP receives more testing before deploying on your own network. Even then, the complexity of the configuration and troubleshooting difficulty might prevent this from being a viable option.

Interior Gateway Routing Protocol

Interior Gateway Routing Protocol (IGRP) is a proprietary Cisco routing protocol meant to address some of the limitations of RIP. The initial version did not address any of its security limitations, however, because IGRP supports no form of authentication. Like RIP, IGRP should be avoided in security-sensitive environments.

Enhanced Interior Gateway Routing Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) is an extension to IGRP that is also Cisco proprietary. It supports MD5 message authentication.

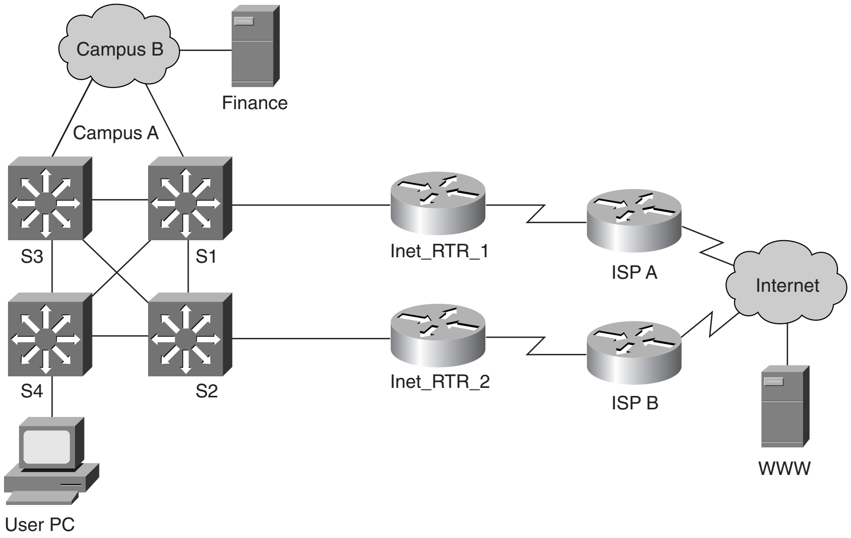
The configuration for EIGRP authentication is as follows:

```
!Specify EIGRP MD5 authentication
Router(config-if)# ip authentication mode eigrp autonomous-system md5
!Specify authentication key
Router(config-if)# ip authentication key-chain eigrp autonomous-system
 name-of-chain
!Identify key chain
Router(config)# key chain name-of-chain
!Specify key number
Router(config-keychain)# key number
!Specify actual key
Router(config-keychain-key)# key-string text
```

Asymmetric Routing and State-Aware Security Technology

As networks increase in size, so do the chances that they have asymmetric traffic somewhere within them. *Asymmetric traffic* is traffic that uses a different path for its return than the original path of the request. The topology in Figure 6-21 shows a representative network with several places where asymmetric traffic can occur.

Figure 6-21 *Asymmetric Traffic*

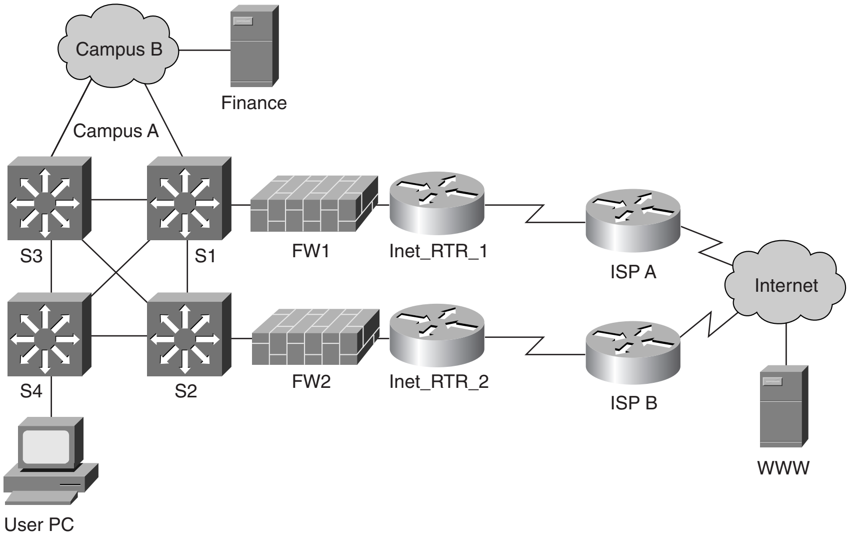


Traffic between the user PC and either the finance server or the WWW server can flow in an asymmetric manner at several points along the network. Between the PC and the finance server, switches S1 and S3 are the main location it can occur. Between the PC and the WWW server, traffic could take an asymmetric route at S1 and S2 or at the Internet when returning through ISP A or ISP B.

So far, this is network design 101. Most network designers don't have any problem with asymmetric traffic because IP networks are asymmetric by nature. At each point in the transmission, an IP router makes a forwarding decision based on its view of the network.

This becomes problematic when security devices are introduced that rely on state information to make forwarding decisions. Consider the revised diagram in Figure 6-22, where two stateful firewalls are introduced between campus A and the two Internet connections.

Figure 6-22 *Asymmetric Traffic with Security Devices*



Now asymmetric flows really start to cause problems! Again, consider the PC communicating with server WWW. A perfectly reasonable packet flow might have the outgoing connection flow through S4, S1, FW1, Inet_RTR_1, ISP A, and then to server WWW. Along the way, FW1 learns that the PC is trying to communicate with server WWW, and so it adds an entry in its state table to enable the return traffic to flow when it comes back from server WWW. Unfortunately, the return path for the packet from server WWW to the user PC happens to be ISP B, Inet_RTR_2, FW2, S2, S4, user PC. The packet never reaches the PC, though, because FW2 doesn't have any state information for the communication. As far as it is concerned, server WWW is initiating new communications to the user PC that are blocked based on the configured security policy.

This problem can be further complicated by intrusion detection systems (IDS) deployed within the campus or near the firewalls. If traffic flows by an IDS in an asymmetric manner, it won't see all of the data. Consequently, it might alarm on traffic that is benign (false positive), or it might miss an attack altogether (false negative).

I wish there were an easy answer to this problem, but unfortunately there isn't. This section is included as much to bring the problem to your attention as it is to offer possible solutions. You do have some options, however:

- Make your routing symmetric.
- Load balance per flow rather than per packet.
- Use state-sharing security devices.
- Consider L2 redundancy as a workaround.
- Manipulate flows by using routing or NAT.
- Use stateless security features.

Make Your Routing Symmetric

This might seem easy, but in real network designs it can be a significant challenge. Even still, you would be surprised to see how many large networks use symmetric routing at certain parts of their network to enable state-aware security devices to function or to solve other networking issues. This is particularly common at Internet edges, where it is not unheard of to see an entire connection to an ISP lying dormant while the primary connection handles all of the load.

Load Balance Per Flow Rather Than Per Packet

Most L3 devices can be configured to do one of two things when equal-cost paths exist for a given network destination. In the first option, packets are simply balanced in round-robin format, with each successive packet going to the next available upstream router. This option causes the most heartache with internal security systems such as IDS. The second, more preferred, option is to load balance based on a given flow. This means traffic with a particular source and destination IP address and port (often called a *four tuple*) is always sent by a specific upstream router. This allows IDS systems and other state-aware devices to at least see half of the communication in a consistent manner. Unfortunately, this does nothing to the return traffic, which still might flow over a different link.

Use State-Sharing Security Devices

As the problem of asymmetric traffic manifests itself more and more in networks, network security vendors are starting to offer options allowing the state information within one security device to be shared with another. In Figure 6-22, FWs 1 and 2 could exchange their state table information to ensure that if the other device sees part of a given flow, it will know to permit the traffic. Often, the amount of information exchanged is significant and requires that dedicated links be configured between the firewalls to exchange the state information.

Consider L2 Redundancy as a Workaround

With the careful introduction of L2 redundancy as opposed to L3, technologies such as Virtual Router Redundancy Protocol (VRRP) or Hot Standby Router Protocol (HSRP) can allow traffic to flow through a single location while still providing redundancy. This option works best on high-speed connections where the use of only one path instead of two or more does not affect network performance.

The result is that normally asymmetric flows can be made symmetric for short distances in the network, such as while traffic passes through a firewall. Again, in Figure 6-22, if FWs 1 and 2 were connected on both sides to the same L2 network, they could use something like VRRP to appear as a single firewall to the upstream and downstream routers. This means that traffic can flow in an asymmetric manner out to the Internet and to the internal network but in a symmetric manner when passing through the firewall. This is generally impossible when the two devices are not in close geographic proximity to one another. For example, if FW 1 is in Brussels, Belgium, and FW 2 is in Hackensack, New Jersey, you are out of luck.

Manipulate Flows by Using Routing or NAT

Because this is a book on security, the ins and outs of BGP path preference have no place within the text. It is worth noting, however, that there are a number of things that can be done with routing protocols to affect the paths that packets take. To some degree, you can also influence which path outside networks take when they must communicate with you. Although not very elegant, some other workarounds involve using different NAT pools based on which security device a packet passes through. Return packets can then be forced to a specific security device based on the unique NAT pool they allocate from.

Use Stateless Security Features

Even though firewalls have been around for many years, a number of companies still use basic ACLs instead of stateful firewalls for, among other things, this asymmetric issue. Some security functionality is clearly lost. Basic ACLs don't track state information, but if your traffic flows are fairly easy to categorize, you can still achieve some security without needing symmetric traffic flows. Remember that if you have properly implemented a true *security system* as defined in Chapter 1, the access control function of a firewall is only one part of the overall security story.

With IDS, the signatures that work improperly in asymmetric environments can be turned off to prevent false positives. Again, this will reduce the security such systems provide but will still allow a number of signatures to fire properly.

Transport Protocol Design Considerations

At the transport level, you often don't have many choices. Applications you use on your network will generally use TCP or UDP. Because TCP is connection oriented and reliable, it is generally preferable to UDP from a security perspective. Keep in mind that UDP is often faster because of the overhead TCP adds.

One of the most significant reasons TCP is more secure than UDP is the difficulty in spoofing TCP communications. As you learned in Chapter 3, UDP spoofing is trivial since there is no notion of connection. This is a main reason why UDP protocols such as SNMP, TFTP, and syslog need special attention when deployed in a security-sensitive environment. Spoofing TCP SYN packets is also easy because no response is needed by the host. (The connection hasn't been formed at this point.) Trying to hijack an established TCP session, however, is very difficult if the attacker is unable to see the packets flow on the wire. This is because the 32-bit sequence number must be guessed by the attacker. More details on UDP and TCP spoofing (including header diagrams) can be found in the "Spoof" section of Chapter 3.

NOTE In the past, initial sequence numbers (ISNs) were not sufficiently random. (In the Kevin Mitnick attack against Tsutomu Shimomura's computers, the ISN incremented by 128,000 for each new session.) Today, however, most modern systems choose much better ISNs that are difficult for an attacker to guess.

DoS Design Considerations

Designing your network to properly deal with DoS/flood attacks is an exercise in compromises. DoS attacks cannot be completely stopped. Anyone who tries to sell you something to completely solve your DoS problems is lying, period. DoS attacks are so easy to launch that they are often considered bad form in the attack community. Three DoS attacks were highlighted in Chapter 3: smurf, DDoS, and TCP SYN. The first two fall into the category of network flooding attacks designed to consume all available bandwidth. The latter is a transport flooding attack designed to consume the resources of a host.

Network Flooding Design Considerations

Detecting a network flood is fairly easy: NIDS, routers, and firewalls can all show signs of a network flood in the log data. *Stopping* a network flood is something entirely different. The chief problem with stopping a network flooding attack is that, by the time the attack reaches your organization, it is already too late. As you learned in Chapter 3, if an attacker sends your T1 connection a T3's worth of data, it doesn't matter if you drop all these packets when they hit your WAN router. Your T1 is already filled, damage done. So, who can help? Your service provider (SP) can.

Your SP has a few specific technologies available, but be aware that most will stop good and bad traffic from reaching the IP address under attack. As an ISP customer, you should have a plan to deal with this eventuality. It should include answers to the following questions:

- How fast can your DNS infrastructure propagate a new IP address for the DNS name under attack?
- Do you currently have redundant systems to which you can make a simple cutover instead of losing legitimate flows?
- What happens if the IP address under attack is your primary router port or other critical infrastructure device? Do you have contingency plans to deal with this?

Additionally, you should know whether your SP offers several of the attack mitigation capabilities outlined in the next sections and, if so, how and when they will be implemented.

Stopping Network Flooding

If you and the ISP decide to just stop the attack outright, there are three primary options, all of which stop good and bad traffic from reaching the victim IP address: basic ACL, black hole filtering, and sinkhole routing.

Basic ACL

The simplest way to stop an attack against a particular IP address is to drop any traffic destined for that IP address. By configuring these ACLs throughout an ISP's network in response to an attack, all traffic destined for the victim IP address can be dropped, stopping the attack. This is a time-consuming process and is the least effective of the three methods.

Black Hole Filtering

Through the clever propagation of static routes in BGP, it is possible to inject a route into the ISP network, causing any traffic destined for the IP that is under attack to be dropped. Traffic is typically routed to null0 (the bit bucket) because this has less CPU impact than dropping the traffic by an ACL (in addition to being much faster to propagate to all ISP routers). Black hole filtering can also be made available to you as an ISP customer if your ISP allows it; see http://www.secsup.org/CustomerBlackHole/ for more information.

Sinkhole Routing

If the ISP is interested instead in examining the flooding attack and stopping it, it can use sinkhole routing. This works by injecting a more specific route from one of the ISP's routers than the subnet route you advertise, which is under attack. For example, if your subnet is 192.0.2.0/24 and IP address 192.0.2.52 is under attack, the ISP can inject a route specifically to the

192.0.2.52/32 address that redirects the attack traffic to a network honeypot of sorts, where the ISP can examine and classify the traffic.

DDoS Trace Back

If, instead, the ISP wants to trace back the source of the attack, there are separate methods to do this. Be aware that trace back is simple if the attack is not spoofing its source address. If the attack uses spoofed source addresses, one of the two primary techniques is used: manual ACL trace back or backscatter DDoS trace back.

Manual ACL Trace Back

When an ISP first tries to categorize an attack, an ACL can be built with a series of broad permit statements that are made more specific as more information about the attack is learned. By measuring the amount of "hits" each ACL entry gets, the ISP is able to determine the kind of traffic that is causing the attack. Once the attack type is determined, a small sampling of traffic can be logged with the **log-input** ACL flag discussed in Chapter 5. This allows the source interface and source MAC address to be determined. By using this information, the ISP can repeat this process on the router that is sourcing the attack. This trace back technique can take time and often results in attack sources on different links in the event of a DDoS flood. Each of these must be traced back individually.

Backscatter DDoS Trace Back

This technique was developed by Chris Morrow and Brian Gemberling at UUNET, and it allows a DDoS attack to be stopped and trace back to occur in approximately 10 minutes. The following site provides more information: http://www.secsup.org/Tracking/.

At a high level, the mitigation technique works by combining aspects of the sinkhole and black hole routing discussed previously. When a system is under attack, the black hole routing technique allows ISP edge routers to route the traffic to null0. This causes an ICMP unreachable to be generated by the router for each spoofed source address that is routed to null. Here's where the trick comes in.

The IPv4 address space is only partially allocated; currently no one owns large blocks of addresses. The list can be found here: http://www.iana.org/assignments/ipv4-address-space. Your ISP can advertise these prefixes and set them to not be exportable to other ISPs. ISPs do this by using the sinkhole router. Because these large blocks of IPv4 address space are now routable within the ISP, all of the ICMP unreachables from spoofed sources in the range the ISP is falsely advertising flow to the sinkhole router. The sinkhole router sees these ICMP unreachables with a source IP address of the router that sent them. Then the ISP has a list of the routers that are seeing the flood attack!

I continue to be amazed by this technique of backscatter trace back. I am constantly surprised by the ingenuity of the Internet's users, and I enjoy it even more when the novel idea is for the cause of good rather than evil. As a note, attackers wishing to get around this method of trace back need only ensure that the spoofed source addresses they use are allocated to legitimate networks.

All of the techniques described in the preceding two subsections are detailed in a North American Network Operators Group (NANOG) presentation from NANOG23 titled "ISP Security—Real World Techniques," delivered by Barry Greene, Chris Morrow, and Brian Gemberling. It is available at the following URL: http://www.nanog.org/mtg-0110/greene.html.

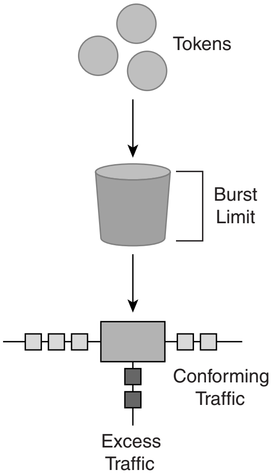
CAR

This DDoS mitigation technique is losing favor because more and more attacks fail to be adequately classified by this technology. CAR is a QoS technique that, for the purposes of flooding mitigation, limits traffic matching an extended ACL to a specific rate. For example, you could use CAR to limit the following types of traffic:

- ICMP traffic to 100 Kbps
- UDP traffic to 5 Mbps
- TCP SYN packets to 50 Kbps

To understand how CAR works, it is helpful to use a common QoS metaphor. CAR works as a token bucket QoS implementation (see Figure 6-23). *Token bucket* means traffic requires a token to pass through the router. Tokens are made available to the limited traffic at the committed rate. If traffic is sent constantly at the committed rate, tokens are constantly spent to pass the traffic. If the rate drops below the committed rate, these tokens can accumulate in a token bucket. The depth of this bucket is equal to the burst rate defined in the CAR statement. If traffic has been below the committed rate for some time, the token bucket will be full. If traffic suddenly exceeds the committed rate for a short period of time, the extra tokens in the bucket can allow the traffic to pass. When the token bucket is completely exhausted, the router is able to take a loan out at the extended burst rate. If the tokens for the extended burst are exhausted, traffic is dropped. As the traffic drops below the committed rate, tokens are first used to pay off the loan for the extended burst before they are put into the token bucket.

Figure 6-23 *CAR*



Like the previous network flooding mitigation techniques, CAR must be implemented by your service provider. Since CAR impacts the performance of a router, expect to pay extra to have your ISP run CAR at all times, or you can work out an agreement in which CAR is turned on after you first detect the attack. To configure CAR to implement the three preceding examples, you start by defining the traffic types by ACLs, as shown in the following example. **permit** means the traffic should be rate limited; **deny** means it should be passed unmolested.

```
! ACL for ICMP Traffic
access-list 102 permit icmp any any
! ACL for TCP SYN Traffic
access-list 103 permit tcp any any syn
! ACL for UDP Traffic
access-list 104 permit udp any any
```

After the ACLs are defined, the rate-limit statements are applied to each ACL. The rate-limit statements can be applied inbound or outbound; because these statements are generally made from the SP's perspective, they are all outbound. After the access list to match is defined, three rates are provided. The first is the committed rate; in the case of ICMP, this is 100 Kbps. The next two numbers are the burst rate and the extended burst rate. The final statements define what the router should do when traffic conforms to the committed rate and what should be done when it exceeds the committed rate. In most cases, the conform action is transmit and the exceed action is drop. Here is what the commands look like:

```
Router(config)#interface S0
Router(config-if)#rate-limit output access-group 102 100000 8000 8000
conform-action transmit exceed-action drop
Router(config-if)#rate-limit output access-group 103 50000 4000 4000
conform-action transmit exceed-action drop
Router(config-if)#rate-limit output access-group 104 5000000 50000 50000
conform-action transmit exceed-action drop
```

CAR Design Considerations

One of the first tasks in successfully configuring CAR is determining what normal traffic loads are. One of the easiest ways to do this is to start your CAR policy by setting your conform action to transmit and your exceed action to transmit. This command for the previous ICMP example looks like this:

```
Router(config-if)#rate-limit output access-group 102 100000 8000 8000
 conform-action transmit exceed-action transmit
```

In this way, no traffic is dropped, but the CAR process is still running. You can then check to see what amount of your traffic is conforming and what is exceeding with the **show interface** *int* **rate-limit** command. The following is an example of the output of this command:

```
Router#sho interface fa0/0 rate-limit
FastEthernet0/0
 Output
  matches: access-group 102
   params: 96000 bps, 8000 limit, 8000 extended limit
   conformed 393 packets, 566706 bytes; action: transmit
   exceeded 4224 packets, 6091008 bytes; action: drop
   last packet: 0ms ago, current burst: 7072 bytes
   last cleared 00:03:51 ago, conformed 19000 bps, exceeded 210000 bps
  matches: access-group 103
   params: 48000 bps, 4000 limit, 4000 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 79586392ms ago, current burst: 0 bytes
   last cleared 00:03:20 ago, conformed 0 bps, exceeded 0 bps
  matches: access-group 104
   params: 48000 bps, 5000 limit, 5000 extended limit
   conformed 0 packets, 0 bytes; action: transmit
   exceeded 0 packets, 0 bytes; action: drop
   last packet: 79586392ms ago, current burst: 0 bytes
   last cleared 00:02:42 ago, conformed 0 bps, exceeded 0 bps
```

From the output, you can see that the router is currently rate limiting a small ICMP flood. You can see the number of packets that exceeded the rate, as well as a number of other interesting statistics.

CAR is powerful because if the attack can be classified properly, the network under attack is unaffected and can continue to service legitimate requests. In the previous ICMP example, a 100 Mbps flood of ICMP traffic would be reduced to a 100 Kbps stream, certainly not enough to adversely affect the network. The main problem with CAR is that it cannot effectively identify certain types of flooding attacks. UDP floods and ICMP floods are easy, but what if you are flooded with TCP 80 traffic with the acknowledgment (ACK) bit set in the TCP header? This is exactly the sort of traffic you should be permitting into the network, so distinguishing the attack by using CAR is impossible.

Also keep in mind that some types of CAR filtering require quite a bit of care in deploying. The TCP SYN option is the most sensitive. Assume that your normal TCP SYN rate is 100 Kbps, and you occasionally spike to 300 Kbps. You implement CAR for TCP SYN and provide a committed rate of 500 Kbps. A TCP SYN flood attack is launched against your network, sending 100 Mbps of TCP SYN traffic—enough to fill up your T3 without CAR. By using CAR, you see only 500 Kbps of the attack, but now any new TCP session won't establish because TCP SYN traffic is being rate limited so extensively. Existing TCP traffic will still pass, but if most are short-lived HTTP connections, the user's web session will quickly stop functioning.

At this point, sessions to that IP address are being dropped, but at least the rest of your network is still functioning. Other systems, routing protocols and so on, continue to work. For this reason, most users choose not to implement TCP SYN flood protection, or any CAR, all the time. Rather, they wait until the attack begins and then work with their ISP to implement the feature.

Design Techniques to Mitigate DDoS

As a security architect, there are two primary techniques you can use to reduce the chances of a successful DDoS attack in the first place: e-commerce-specific filtering and content delivery networks.

E-Commerce-Specific Filtering

In most designs, the e-commerce portion of an organization's network uses the same bandwidth as the rest of the network. Users, mail servers, and e-commerce transactions all occur over the same WAN link. This is suboptimal for several reasons:

- A successful flood attack against your Internet connection will affect both general Internet and e-commerce traffic.
- A spike in internal user Internet usage can affect e-commerce availability.
- Because internal user traffic is so diverse (lots of applications, ports, and protocols), the usage of the WAN link can be unpredictable.

Instead, organizations could choose to separate their internal users from their e-commerce systems in one of two ways:

Move the e-commerce environment to a collocation facility at your SP, as shown in Figure 6-24.
Purchase two separate Internet connections (four if you need redundancy for both services), as in Figure 6-25.

Figure 6-24 *Collocated E-Commerce*

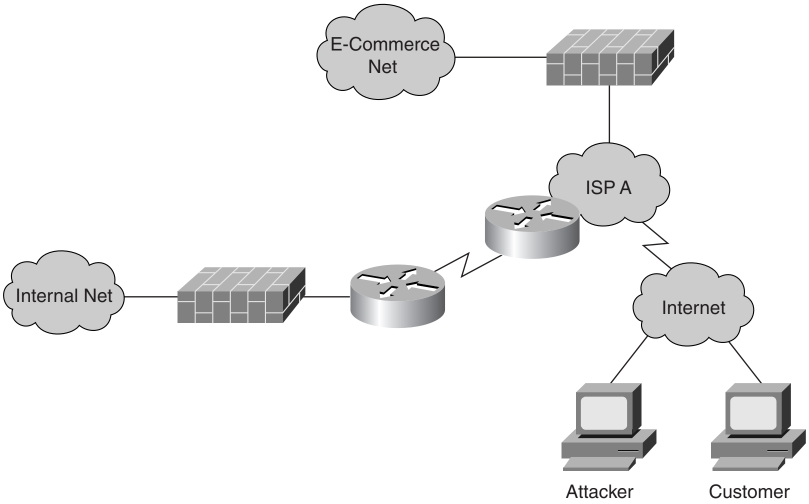
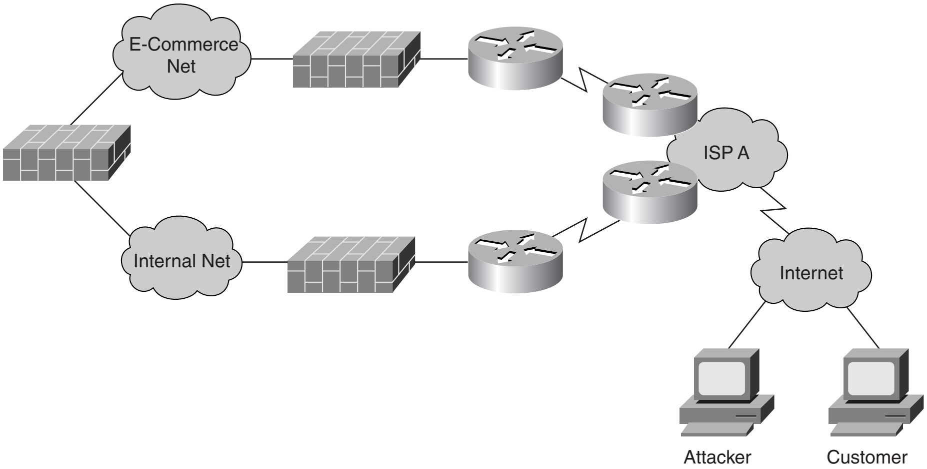


Figure 6-25 *Dedicated E-Commerce WAN Connection*



In the collocation example, you have the benefit of increased bandwidth because you are physically sitting within the ISP's network, whereas in the second example you have greater control and manageability of your e-commerce systems. In either case, specific filtering works the same. In an e-commerce environment, you typically need a very limited set of services to function, including the following:

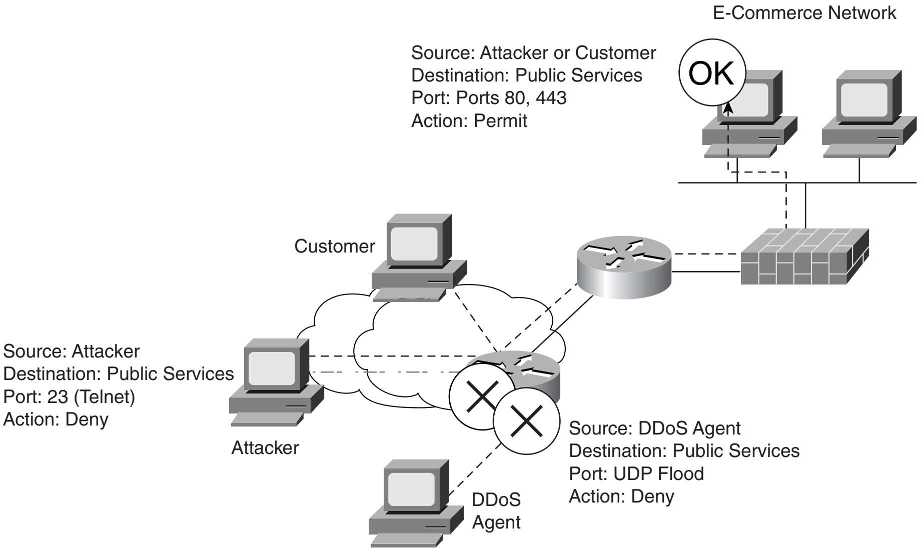
- HTTP (TCP port 80)
- SSL/Transport Layer Security (TLS) (TCP port 443)
- BGP (TCP port 179)
- ICMP (as defined earlier in this chapter)

DNS is not needed if the DNS servers are hosted somewhere else, such as at the ISP. This means UDP as a whole may not be needed. With this level of specificity, it becomes possible to filter e-commerce traffic as it leaves the ISP network destined for your e-commerce systems. This provides two distinct advantages:

- DDoS or worms must be very specific in order to reach the e-commerce network.
- Traffic that would otherwise consume expensive ISP bandwidth can be stopped. Because the traffic is blocked at the firewall anyway, there is no sense in allowing it on the wire in the first place.

E-commerce-specific filtering is shown in Figure 6-26.

Figure 6-26 *E-Commerce-Specific Filtering*



Don't think of this as a service-provider-managed firewall; all you are asking your SP to do is implement a basic ACL outbound on your interface. If your BGP router IP is 96.20.20.2, the SP router IP is 96.20.20.1, and your web/SSL server is 192.0.2.50, the ACL would look like this:

```
Router(config)#access-list 101 permit tcp any gt 1023 host 192.0.2.50 eq 80
Router(config)#access-list 101 permit tcp any gt 1023 host 192.0.2.50 eq 443
Router(config)#access-list 101 permit icmp any any
Router(config)#interface s0
Router(config-if)#ip access-group 101 out
```

NOTE Notice that the BGP traffic did not need to be permitted by the ACL since traffic originated by the router is not filtered in an ACL. You should make the ICMP filtering more specific as discussed earlier in the chapter. This kind of filtering should also be combined with RFC 2827 filtering and bogon filtering as well.

Content Delivery Networks

The second design option is to distribute your critical systems in multiple data centers using network load balancing to distribute the content. This doesn't stop a DDoS attack, but it does lessen its significance because the other systems serving the same content are still online. Content delivery networks are touched on in Chapter 11, "Supporting-Technology Design Considerations."

Network Flooding Design Recommendations

All of the technologies described in this section can be considered potential tools in your network flooding toolkit. All of them require cooperation with your ISP, something that should be put in writing *before* you are attacked. Most ISPs should be receptive to cooperating on a network flooding policy, particularly if they don't yet have your business. Be sure to discuss such options as the methods that will be used for different attacks as well as how quickly you can expect a turnaround when an attack occurs. Understanding the relationships your ISP has with other ISPs can also be helpful in understanding how well it will respond when attacks occur.

TCP SYN Flooding Design Considerations

TCP SYN flooding was discussed in Chapter 3. Stopping such attacks can be done either at the host only or at the host in combination with the network. The two principal technologies to mitigate SYN flooding are SYN cookies and TCP Intercept.

SYN Cookies

SYN cookies are a host-specific method of mitigating TCP SYN flooding attacks. When the incoming SYN queue fills up from attack, a server normally must block new incoming connections. When using SYN cookies, instead of keeping each SYN in the queue, information from the SYN sent from the client is run through a cryptographic function to determine the ISN to send from the server. This way, the server mustn't keep track of the SYN packet; it must only check an incoming ACK for a new session against this cryptographic function. The ACK from the client should be exactly one more than the ISN sent from the server. By decrypting this value, the server has enough essential information to allow the TCP connection to establish, even without a copy of the original SYN. More information on SYN cookies can be found at http://cr.yp.to/syncookies.html.

TCP Intercept

TCP Intercept is a network-level protection for SYN floods. It works by brokering (on the device running TCP Intercept) a connection to a server on behalf of the client. If an incoming connection never establishes itself, the client is not affected. When the connection does establish, the device running TCP intercept passes the communication on to the real server transparently. The Cisco PIX documentation does a good job of describing the feature in detail, so I've included it here. An *embryonic connection* in Cisco terminology is one that has not completed the full TCP three-way handshake:

> Once the optional embryonic connection limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the affected server is intercepted. For each SYN, PIX Firewall responds on behalf of the server with an empty SYN/ACK segment. PIX Firewall retains pertinent state information, drops the packet, and waits for the client's acknowledgement. If the ACK is received, a copy of the client's SYN segment is sent to the server, and the TCP three-way handshake is performed between PIX Firewall and the server. If, and only if, this three-way handshake completes, may the connection can resume as normal. If the client does not respond during any part of the connection phase, PIX Firewall retransmits the necessary segment using exponential back-offs.

TCP Intercept has a number of options when implemented on routers; for more information, see the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfdenl.htm. On the Cisco PIX Firewall, TCP Intercept is part of the **static** command and has only one configurable option: the number of half-open connections to accept before starting the intercept function. More information on the **static** command can be found at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/cmdref/s.htm#1026694.

ICMP Unreachable DoS Considerations

If a request comes in to the router directed to a service the router isn't running, an ICMP unreachable message is sent. Sending ICMP unreachables could be used to deny service of the router. If an attacker can keep the router sending unreachables, the overall service of the router could degrade. To silently discard these packets without generating a message, the following command should be configured on each interface:

```
Router(config-if)#no ip unreachables
```

Earlier in this section, a DDoS traceback technique is used, which involves ICMP unreachables. Additionally, path MTU discovery uses ICMP unreachable messages, so blocking it will stop path MTU, which often isn't a good thing. If you need unreachables for this or any other reason, consider rate limiting them with the following command instead of dropping them outright:

```
Router(config)#ip icmp rate-limit unreachable milliseconds
```

This will prevent the router from being consumed with the process of sending ICMP unreachables.

Summary

This chapter presents a lot of information on best practices, covering everything from physical security to DoS design considerations. Many of the best practices are not implemented on special-purpose security gear but rather on the networking gear you probably already have in place. In Chapter 4, you saw how special-purpose security technology could impact the threats discussed in Chapter 3. In Chapter 5, you learned hardening practices for network elements and hosts. At this point, it is appropriate to revisit the table at the end of Chapter 4 that shows the threats and the technologies that help detect or stop them. Table 6-1 shows this information again with the technology and techniques from Chapters 5 and 6 integrated.

Table 6-1 *Threat Mitigation Summary*

		Attack Class	Read					Manipulate			Spoof					
		Attack Subclass	*Reconnaissance*						*Application Manipulation*				*Transport Spoofing*			
D = Detect S = Stop		Attack Element	Data Scavenging	Probe/ Scan	War Dialing/ Driving	Sniffer	Direct Access	Network Manipulation	Buffer Overflow	Web Application	MAC Spoofing	IP Spoofing	UDP Spoofing	TCP Spoofing	Identity Spoofing	Rogue Devices
Technology Class	**Technology**	Overall Score														
Identity	OTP	63					**S**								**S**	
	RADIUS / TACACS+	61					**S**								**D**	
	Reusable Passwords	59					**S**								**D**	
	Smart Cards	55					**S**								**S**	
	PKI	54					**S**								**S**	
	Biometrics	52					**S**								**S**	
Host and App	Host AV	66														
	FS Check	61							**D**	**D**						
	HIDS	61		**D**			**D**		**D**	**D**						
	Host Firewalls	51		**D**			**S**									
Network FW	Stateful FW	89					**S**	**S**				**S**				
	Router with ACL	80					**S**	**S**				**S**				
Content	E-mail Filtering	69														
	Web Filtering	53					**S**									
	Proxy Server	43					**S**									
NIDS	Signature-Based NIDS	68		**D**				**D**	**D**	**D**		**D**				
	Anomaly-Based NIDS	51														
Crypto	Network Crypto	96				**S**	**S**					**S**			**S**	
	FS Crypto	91					**S**								**S**	
	L2 Crypto	91				**S**	**S**				**S**				**S**	
	L5 - L7 Crypto	88				**S**	**S**								**S**	
Hardening	Network Device Hardening			**S**			**S**									
	OS Hardening			**D**			**S**		**S**							
	Application Hardening			**D**			**S**		**S**	**S**						
	Rogue Device Detection				**S**											**S**
Best Practices (BP)	Physical Security				**S**		**S**									
Layer 2 BP	L2 Control Protocol BPs					**S**										
	Port Security					**S**					**S**					
	VLAN Hopping BPs						**S**									
	ARP BPs					**S**										
	DHCP BPs					**S**										
	Private VLANs					**S**	**S**									
Layer 3 BP	Role-Based Subnetting						**S**									
	Ingress / Egress Filtering											**S**	**S**	**S**		
	Unicast RPF											**S**	**S**	**S**		
	ICMP BPs			**S**												
	Routing Protocol Auth						**S**									**S**
DoS BPs	DDoS BPs															
	TCP SYN BPs															

continues

Table 6-1 *Threat Mitigation Summary (Continued)*

		Attack Class	Flood					Redirect				Composite			
		Attack Subclass		*Network Flooding*				*L2 Redirection*							
D = Detect S = Stop		Attack Element	MAC Flooding	Smurf	Distributed Denial of Service (DDoS)	TCP SYN Flood	Application Flooding	ARP redirection/ spoofing	STP Redirection	IP Redirection	Transport Redirection	Man-in-the-Middle (MITM)	Virus/Worm/ Trojan Horse	Rootkit	Remote Control Software
Technology Class	**Technology**	Overall Score													
Identity	OTP	63													
	RADIUS / TACACS+	61													
	Reusable Passwords	59													
	Smart Cards	55													
	PKI	55													
	Biometrics	52													
Host and App	Host AV	66											**S**		**S**
	FS Check	61											**D**	**D**	
	HIDS	61				**D**					**D**				**D**
	Host Firewalls	51													**S**
Network FW	Stateful FW	89		**D**	**D**	**S**									
	Router with ACL	80		**D**	**D**					**S**					
Content	E-mail Filtering	69											**S**		**S**
	Web Filtering	53											**S**		
	Proxy Server	43													
NIDS	Signature-Based NIDS	68		**D**	**D**	**D**		**D**					**D**		**D**
	Anomaly-Based NIDS	51		**D**	**D**	**D**							**D**		
Crypto	Network Crypto	96										**S**			
	FS Crypto	91												**S**	**S**
	L2 Crypto	91										**S**			
	L5 - L7 Crypto	88										**S**			
Hardening	Network Device Hardening			**S**											
	OS Hardening										**S**			**S**	**S**
	Application Hardening										**S**			**S**	**S**
	Rogue Device Detection											**S**			
Best Practices (BP)	Physical Security														
Layer 2 BP	L2 Control Protocol BPs								**S**						
	Port Security		**S**												
	VLAN Hopping BPs														
	ARP BPs							**S**				**S**			
	DHCP BPs											**S**			
	Private VLANs							**S**							
Layer 3 BP	Role-Based Subnetting														
	Ingress / Egress Filtering				**S**	**S**									
	Unicast RPF				**S**	**S**									
	ICMP BPs														
	Routing Protocol Auth									**S**					
DoS BPs	DDoS BPs			**S**	**S**								**D**		
	TCP SYN BPs					**S**									

References

- Arkin, O. "ICMP Usage in Scanning." http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.5.pdf
- arpwatch. http://www-nrg.ee.lbl.gov/
- Baker, F., and R. Atkinson. RFC 2082, "RIP-2 MD5 Authentication." http://www.ietf.org/rfc/rfc2082.txt
- Cisco ACL Fragmentation Issues. http://www.cisco.com/warp/public/105/acl_wp.html
- Cisco Documentation: ARP Inspection. http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_7_5/confg_gd/acc_list.htm#1020673
- Cisco Documentation: DHCP Snooping. http://www.cisco.com/univercd/cc/td/doc/product/lan/cat4000/12_1_13/config/dhcp.htm
- Cisco Documentation: PIX Static Command. http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/cmdref/s.htm#1026694
- Cisco Documentation: Port Security. http://cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_5_4/config/sec_port.htm
- Cisco Documentation: Private VLANs. http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_7_1/conf_gd/vlans.htm#xtocid854519
- Cisco Documentation: TCP Intercept. http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/ftrafwl/scfdenl.htm
- Cisco Documentation: Unicast RPF. http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fsecur_c/fothersf/scfrpf.htm
- Convery, S. "Hacking Layer 2: Fun with Ethernet Switches." http://www.blackhat.com/presentations/bh-usa-02/bh-us-02-convery-switches.pdf
- DHCP DoS. http://packetstormsecurity.org/DoS/DHCP_Gobbler.tar.gz
- dsniff. http://monkey.org/~dugsong/dsniff/
- Ferguson, P., and D. Senie. RFC 2827, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing." http://www.ietf.org/rfc/rfc2827.txt
- Greene, B., C. Morrow, and B. Gemberling. "ISP Security—Real World Techniques." http://www.nanog.org/mtg-0110/greene.html
- Heffernan, A. RFC 2385, "Protection of BGP Sessions via the TCP MD5 Signature Option." http://www.ietf.org/rfc/rfc2385.txt
- IANA IPv4 Address Allocation. http://www.iana.org/assignments/ipv4-address-space
- Kuhn, M. and R. Anderson. "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations." http://www.cl.cam.ac.uk/~mgk25/ih98-tempest.pdf

- Kuhn, Markus G. "Optical Time-Domain Eavesdropping Risks of CRT Displays." http://www.cl.cam.ac.uk/~mgk25/ieee02-optical.pdf
- Malkin, G. RFC 1723, "RIP Version 2 Carrying Additional Information." http://www.ietf.org/rfc/rfc1723.txt
- Morrow, C., and B. Gemberling. "Backscatter DDoS Traceback." http://www.secsup.org/Tracking/
- Morrow, C., and B. Gemberling. "Enabling Black Hole Filtering for Customers." http://www.secsup.org/CustomerBlackHole/
- Moy, J. RFC 2328, "OSPF Version 2." http://www.ietf.org/rfc/rfc2328.txt
- Neil Jr. "Spy Agency Taps into Undersea Cable." *Wall Street Journal*. http://zdnet.com.com/2100-11-529826.html
- Ping of Death. http://www.insecure.org/sploits/ping-o-death.html
- Portable Keystroke Logger. http://www.thinkgeek.com/stuff/gadgets/5a05.shtml
- Rekhter, Y., B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. RFC 1918, "Address Allocation for Private Internets." http://www.ietf.org/rfc/rfc1918.txt
- SYN Cookies. http://cr.yp.to/syncookies.html
- Taylor, David. "Are There Vulnerabilities in VLAN Implementations?" http://www.sans.org/resources/idfaq/vlan.php
- Thomas, Rob. "Bogon List." http://www.cymru.com/Documents/bogon-list.html
- Thomas, Rob. "ICMP Filtering Guidelines." http://www.cymru.com/Documents/icmp-messages.html
- VLAN 1 Considerations. http://www.cisco.com/warp/public/473/103.html
- van Eck, Wim. "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?" http://www.shmoo.com/tempest/emr.pdf

Applied Knowledge Questions

The following questions are designed to test your knowledge of general network security design considerations, and they sometimes build on knowledge found elsewhere in the book. You might find that each question has more than one possible answer. The answers provided in Appendix B are intended to reinforce concepts that you can apply in your own networking environment.

1. What would the inbound ACL look like on your router's serial interface connected to the Internet if you decided to block RFC 1918 addresses, the bogons listed in this chapter, and RFC 2827 filtering, assuming your local IP range is 96.0.20.0/24?
2. When evaluating the SYN flood protections required for a server, when might you use SYN cookies and when might you use TCP Intercept?

3 What is the most important step when you are trying to get help from your ISP to stop a DDoS attack?

4 When might it not be necessary to implement L2 security features on your network?

5 Should the average user worry about van Eck phreaking?

6 When should you use uRPF as compared to traditional ACL filtering?

7 Is it worth implementing Rob Thomas's entire bogon-filtering range on your Internet edge?