Network security is a broad topic that can be addressed at the data link, or media, level (where packet snooping and encryption problems can occur), at the network, or protocol, layer (the point at which Internet Protocol (IP) packets and routing updates are controlled), and at the application layer (where, for example, host-level bugs become issues).

As more users access the Internet and as companies expand their networks, the challenge to provide security for internal networks becomes increasingly difficult. Companies must determine which areas of their internal networks they must protect, learn how to restrict user access to these areas, and determine which types of network services they should filter to prevent potential security breaches.

Cisco Systems provides several network, or protocol, layer features to increase security on IP networks. These features include controls to restrict access to routers and communication servers by way of console port, Telnet, Simple Network Management Protocol (SNMP), Terminal Access Controller Access Control System (TACACS), vendor token cards, and access lists. Firewall architecture setup is also discussed.

Caution: Although this case study addresses network-layer security issues, which are the most relevant in the context of an Internet connection, ignoring host-level security, even with network-layer filtering in place, can be dangerous. For host-level security measures, refer to your application's documentation and the recommended reading list at the end of this case study.

Guide Contents
Internetworking Design Basics
Designing various internetworks
Network Enhancements
IP Routing Concepts
UDP Broadcast Flooding
Large-Scale H.323 Network Design for Service Providers
LAN Switching
Subnetting an IP Address Space
IBM Serial Link Implementation Notes
References and Recommended Reading

Contents

- 1 Understanding Cisco's Approach to Network Security
- 2 Controlling Access to Cisco Routers
  - 2.1 Console Access
    - 2.1.1 Nonprivileged Mode Password
    - 2.1.2 Privileged Mode Password
    - 2.1.3 Session Timeouts
    - 2.1.4 Password Encryption
  - 2.2 Telnet Access
    - 2.2.1 Nonprivileged Mode Password
    - 2.2.2 Privileged Mode Password
    - 2.2.3 Restricting Telnet Access to Particular IP Addresses
    - 2.2.4 Restricting Telnet Access to Cisco Products via TCP Ports
      - 2.2.4.1 Earlier Software Releases
      - 2.2.4.2 Table: TCP Port Telnet Access to Cisco Products (Earlier Releases)

- 2.2.4.3 Software Releases 9.1 (11.5), 9.21 (3.2), and 10.0 and Later
- 2.2.4.4 Table: TCP Port Telnet Access to Cisco Products (Later Releases)
- 2.2.5 Terminal Access Controller Access Control System (TACACS)
- 2.2.5.1 Nonprivileged Access
- 2.2.5.2 Privileged Access
- 2.2.5.3 Token Card Access
- 2.3 Simple Network Management Protocol (SNMP) Access
- 2.3.1 Nonprivileged Mode
- 2.3.2 Privileged Mode
- 2.4 Controlling Access to Network Servers That Contain Configuration Files
- 3 Setting Up Your Firewall Architecture
- 3.1 Figure: Typical firewall architecture
- 4 Controlling Traffic Flow
- 4.1 Figure: Controlling traffic flow via the firewall router
- 4.2 Configuring the Firewall Router
- 4.2.1 Defining Access Lists
- 4.2.1.1 File Transfer Protocol (FTP) Port
- 4.2.2 Applying Access Lists to Interfaces
- 4.2.2.1 Filtering TCP and UDP Services
- 4.2.2.2 Table: Well-Known TCP and UDP Services and Ports
- 4.2.2.3 CERT Advisory
- 4.2.2.4 Table: CERT Advisory on TCP and UDP Services and Ports
- 4.2.2.5 Input Access Lists
- 4.2.2.6 Figure: A host that is spoofing
- 4.3 Configuring the Firewall Communication Server
- 4.3.1 Defining Access Lists
- 4.3.2 Applying Access Lists to Lines
- 4.4 Using Banners to Set Up Unauthorized Use Notifications
- 5 Securing Nonstandard Services
- 6 Summary
- 7 Recommended Reading
- 7.1 Requests For Comments (RFCs)
- 7.2 Internet Directories

Understanding Cisco's Approach to Network Security

When most people talk about security, they mean ensuring that users can only perform tasks they are authorized to do, can only obtain information they are authorized to have, and cannot cause damage to the data, applications, or operating environment of a system.

The word security connotes protection against malicious attack by outsiders. Security also involves controlling the effects of errors and equipment failures. Anything that can protect against a deliberate, intelligent, calculated attack will probably prevent random misfortune as well.

Security measures keep people honest in the same way that locks do. This case study provides specific actions you can take to improve the security of your network. Before going into specifics, however, it will help if you understand the following basic concepts that are essential to any security system:

- Know your enemy

  This case study refers to attackers or intruders. Consider who might want to circumvent your security measures and identify their motivations. Determine what they might want to do and the damage that they could cause to your network.
  Security measures can never make it impossible for a user to perform unauthorized tasks with a computer system. They can only make it harder. The goal is to make sure the network security controls are beyond the attacker's ability or motivation.

- Count the cost

  Security measures almost always reduce convenience, especially for sophisticated users. Security can delay work and create expensive administrative and educational overhead. It can use significant computing resources and require dedicated hardware.
  When you design your security measures, understand their costs and weigh those costs against the potential benefits. To do that, you must understand the costs of the measures themselves and the costs and likelihoods of security breaches. If you incur security costs out of proportion to the actual dangers, you have done yourself a disservice.

- Identify your assumptions

  Every security system has underlying assumptions. For example, you might assume that your network is not tapped, or that attackers know less than you do, that they are using standard software, or that a locked room is safe. Be sure to examine and justify your assumptions. Any hidden assumption is a potential security hole.

- Control your secrets

  Most security is based on secrets. Passwords and encryption keys, for example, are secrets. Too often, though, the secrets are not really all that secret. The most important part of keeping secrets is knowing the areas you need to protect. What knowledge would enable someone to circumvent your system? You should jealously guard that knowledge and assume that everything else is known to your adversaries. The more secrets you have, the harder it will be to keep all of them. Security systems should be designed so that only a limited number of secrets need to be kept.

- Remember human factors

Many security procedures fail because their designers do not consider how users will react to them. For example, because they can be difficult to remember, automatically generated "nonsense" passwords are often found written on the undersides of keyboards. For convenience, a "secure" door that leads to the system's only tape drive is sometimes propped open. For expediency, unauthorized modems are often connected to a network to avoid onerous dial-in security measures.

If your security measures interfere with essential use of the system, those measures will be resisted and perhaps circumvented. To win compliance, you must make sure that users can get their work done, and you must sell your security measures to users. Users must understand and accept the need for security.

Any user can compromise system security, at least to some degree. Passwords, for instance, can often be found simply by calling legitimate users on the telephone, claiming to be a system administrator, and asking for them. If your users understand security issues, and if they understand the reasons for your security measures, they are far less likely to make an intruder's life easier.

At a minimum, users should be taught never to release passwords or other secrets over unsecured telephone lines (especially cellular telephones) or electronic mail (email). Users should be wary of questions asked by people who call them on the telephone. Some companies have implemented formalized network security training for their employees; that is, employees are not allowed access to the Internet until they have completed a formal training program.

- Know your weaknesses

  Every security system has vulnerabilities. You should understand your system's weak points and know how they could be exploited. You should also know the areas that present the largest danger and prevent access to them immediately. Understanding the weak points is the first step toward turning them into secure areas.

- Limit the scope of access

  You should create appropriate barriers inside your system so that if intruders access one part of the system, they do not automatically have access to the rest of the system. The security of a system is only as good as the weakest security level of any single host in the system.

- Understand your environment

  Understanding how your system normally functions, knowing what is expected and what is unexpected, and being familiar with how devices are usually used, help you to detect security problems. Noticing unusual events can help you to catch intruders before they can damage the system. Auditing tools can help you to detect those unusual events.

- Limit your trust

  You should know exactly which software you rely on, and your security system should not have to rely upon the assumption that all software is bug-free.

- Remember physical security

  Physical access to a computer (or a router) usually gives a sufficiently sophisticated user total control over that computer. Physical access to a network link usually allows a person to tap that link, jam it, or inject traffic into it. It makes no sense to install complicated software security measures when access to the hardware is not controlled.

- Security is pervasive

  Almost any change you make in your system may have security effects. This is especially true when new services are created. Administrators, programmers, and users should consider the security implications of every change they make. Understanding the security implications of a change is something that takes practice. It requires lateral thinking and a willingness to explore every way in which a service could potentially be manipulated.

Controlling Access to Cisco Routers

It is important to control access to your Cisco routers. You can control access to the router using the following methods:

- Console Access
- Telnet Access

- Simple Network Management Protocol (SNMP) Access
- Controlling Access to Network Servers That Contain Configuration Files

You can secure the first three of these methods by employing features within the router software. For each method, you can permit nonprivileged access and privileged access for a user (or group of users). Nonprivileged access allows users to monitor the router, but not to configure the router. Privileged access allows the user to fully configure the router.

For console port and Telnet access, you can set up two types of passwords. The first type of password, the login password, allows the user nonprivileged access to the router. After accessing the router, the user can enter privileged mode by entering the **enable** command and the proper password. Privileged mode provides the user with full configuration capabilities.

SNMP access allows you to set up different SNMP community strings for both nonprivileged and privileged access. Nonprivileged access allows users on a host to send the router SNMP get-request and SNMP get-next-request messages. These messages are used for gathering statistics from the router. Privileged access allows users on a host to send the router SNMP set-request messages in order to make changes to the router's configurations and operational state.

Console Access

A console is a terminal attached directly to the router via the console port. Security is applied to the console by asking users to authenticate themselves via passwords. By default, there are no passwords associated with console access.

Nonprivileged Mode Password

You configure a password for nonprivileged mode by entering the following commands in the router's configuration file. Passwords are case-sensitive. In this example, the password is "1forAll."

```
line console 0
login
password 1forAll
```

When you log in to the router, the router login prompt is as follows:

```
User Access Verification
Password:
```

You must enter the password "1forAll" to gain nonprivileged access to the router. The router response is as follows:

```
router>
```

Nonprivileged mode is signified on the router by the > prompt. At this point, you can enter a variety of commands to view statistics on the router, but you cannot change the configuration of the router. Never use "cisco," or other obvious derivatives, such as "pancho," for a Cisco router password. These will be the first passwords intruders will try if they recognize the Cisco login prompt.

Privileged Mode Password

Configure a password for privileged mode by entering the following commands in the router's configuration file. In this example, the password is "san-fran."

```
enable-password san-fran
```

To access privileged mode, enter the following command:

```
router> enable
Password:
```

Enter the password "san-fran" to gain privileged access to the router. The router responds as follows:

```
router#
```

Privileged mode is signified by the # prompt. In privileged mode, you can enter all commands to view statistics and configure the router.

Session Timeouts

Setting the login and enable passwords may not provide enough security in some cases. The timeout for an unattended console (by default 10 minutes) provides an additional security measure. If the console is left unattended in privileged mode, any user can modify the router's configuration. You can change the login timeout via the command **exec-timeout** mm ss, where mm is minutes and ss is seconds. The following commands change the timeout to 1 minute and 30 seconds:

```
line console 0
exec-timeout 1 30
```

Password Encryption

All passwords on the router are visible via the **write terminal** and **show configuration** privileged mode commands. If you have access to privileged mode on the router, you can view all passwords in cleartext by default.

There is a way to hide cleartext passwords. The command **service password-encryption** stores passwords in an encrypted manner so that anyone performing a **write terminal** and **show configuration** will not be able to determine the cleartext password. However, if you forget the password, regaining access to the router requires you to have physical access to the router.

Note: Although encryption is helpful, it can be compromised and thus should not be your only network-security strategy.

Telnet Access

You can access both nonprivileged and privileged mode on the router via Telnet. As with the console port, Telnet security is provided when users are prompted by the router to authenticate themselves via passwords.

In fact, many of the same concepts described in the "Console Access" section earlier in this article apply to Telnet access. You must enter a password to go from nonprivileged mode to privileged mode, and you can encrypt passwords and specify timeouts for each Telnet session.

Nonprivileged Mode Password

Each Telnet port on the router is known as a virtual terminal. There are a maximum of five virtual terminal (VTY) ports on the router, allowing five concurrent Telnet sessions. (The communication server provides more VTY ports.) On the router, the virtual terminal ports are numbered from 0 through 4. You can set up nonprivileged passwords for Telnet access via the virtual terminal ports with the following configuration commands. In this example, virtual terminal ports 0 through 4 use the password "marin":

```
line vty 0 4
login
password marin
```

When a user telnets to a router IP address, the router provides a prompt similar to the following:

```
% telnet router
Trying ...
Connected to router.
Escape character is '^]'.
User Access Verification
Password:
```

If the user enters the correct nonprivileged password, the following prompt appears:

```
router>
```

Privileged Mode Password

The user now has nonprivileged access to the router and can enter privileged mode by entering the **enable** command as described in the "Privileged Mode Password" section earlier in this article.

Restricting Telnet Access to Particular IP Addresses

If you want to allow only certain IP addresses to use Telnet to access the router, you must use the **access-class** command. The command **access-class** *nn* **in** defines an access list (from 1 through 99) that allows access to the virtual terminal lines on the router. The following configuration commands allow incoming Telnet access to the router only from hosts on network 192.85.55.0:

```
access-list 12 permit 192.85.55.0 0.0.0.255
line vty 0 4
access-class 12 in
```

Restricting Telnet Access to Cisco Products via TCP Ports

It is possible to access Cisco products via Telnet to specified TCP ports. The type of Telnet access varies, depending upon the following Cisco software releases:

- Software Release 9.1 (11.4) and earlier and 9.21 (3.1) and earlier
- Software Release 9.1 (11.5), 9.21 (3.2), and 10.0 and later

Earlier Software Releases

For Software Release 9.1 (11.4) and earlier and Software Release 9.21 (3.1) and earlier, it is possible, by default, to establish TCP connections to Cisco products via the TCP ports listed in Table: TCP Port Telnet Access to Cisco Products (Earlier Releases).

Table: TCP Port Telnet Access to Cisco Products (Earlier Releases)

TCP Port Number	Access Method
7	Echo
9	Discard
23	Telnet (to virtual terminal VTY ports in rotary fashion)
79	Finger
1993	SNMP over TCP
2001 through 2999	Telnet to auxiliary (AUX) port, terminal (TTY) ports, and virtual terminal (VTY) ports
3001 through 3999	Telnet to rotary ports (access via these ports is only possible if the rotaries have been explicitly configured first with the **rotary** command)
4001 through 4999	Telnet (stream mode) mirror of 2000 range
5001 through 5999	Telnet (stream mode) mirror of 3000 range (access via these ports is possible only if the rotaries have been explicitly configured first)
6001 through 6999	Telnet (binary mode) mirror of 2000 range
7001 through 7999	Telnet (binary mode) mirror of 3000 range (access via these ports is possible only if the rotaries have been explicitly configured first)
8001 through 8999	Xremote (communication servers only)
9001 through 9999	Reverse Xremote (communication servers only)
10001 through 19999	Reverse Xremote rotary (communication servers only; access via these ports is possible only if the ports have been explicitly configured first)

Caution: Because Cisco routers have no TTY lines, configuring access (on communication servers) to terminal ports 2002, 2003, 2004, and greater could potentially provide access (on routers) to virtual terminal lines 2002, 2003, 2004, and greater. To provide access only to TTY ports, you can create access lists to prevent access to VTYs.

When configuring rotary groups, keep in mind that access through any available port in the rotary group is possible (unless access lists are defined). Cisco recommends that if you are using firewalls that allow in-bound TCP connection to high-number ports, remember to apply appropriate in-bound access lists to Cisco products.

The following is an example illustrating an access list denying all in-bound Telnet access to the auxiliary port and allowing Telnet access to the router only from IP address 192.32.6.7:

```
access-class 51 deny 0.0.0.0 255.255.255.255
access-class 52 permit 192.32.6.7
line aux 0
```

```
access-class 51 in
line vty 0 4
access-class 52 in
```

To disable connections to the echo and discard ports, you must disable these services completely with the **no service tcp-small-servers** command.

Caution: If the **ip alias** command is enabled on Cisco products, TCP connections to any destination port are considered valid connections. You may want to disable the **ip alias** command.

You might want to create access lists to prevent access to Cisco products via these TCP ports. For information on how to create access lists for routers, see the "Configuring the Firewall Router" section later in this article. For information on how to create access lists for communication servers, see the "Configuring the Firewall Communication Server" section later in this article.

Software Releases 9.1 (11.5), 9.21 (3.2), and 10.0 and Later

With Software Release 9.1 (11.5), 9.21 (3.2), and any version of Software Release 10, the following enhancements have been implemented:

- Direct access to virtual terminal lines (VTYs) through the 2000, 4000, and 6000 port ranges has been disabled. If you want to keep access open, you can set up one-to-one mapping of VTY-to-rotary ports.
- Connections to echo and discard ports (7 and 9) can be disabled with the **no service tcp-small-servers** command.
- All Cisco products allow connections to IP alias devices only on destination port 23.

For later releases, a Cisco router accepts TCP connections on the ports listed in Table: TCP Port Telnet Access to Cisco Products (Later Releases) by default.

Table: TCP Port Telnet Access to Cisco Products (Later Releases)

TCP Port Number	Access Method
7	Echo
9	Discard
23	Telnet
79	Finger
1993	SNMP over TCP
2001	Auxiliary (AUX) port
4001	Auxiliary (AUX) port (stream)
6001	Auxiliary (AUX) port (binary)

Access via port 23 can be restricted by creating an access list and assigning it to virtual terminal lines. Access via port 79 can be disabled with the **no service finger** command. Access via port 1993 can be controlled with SNMP access lists. Access via ports 2001, 4001, and 6001 can be controlled with an access list placed on the auxiliary port.

Terminal Access Controller Access Control System (TACACS)

Nonprivileged and privileged mode passwords are global and apply to every user accessing the router from either the console port or from a Telnet session. As an alternative, the Terminal Access Controller Access Control System (TACACS) provides a way to validate every user on an individual basis before they can gain

access to the router or communication server. TACACS was derived from the United States Department of Defense and is described in Request For Comments (RFC) 1492. TACACS is used by Cisco to allow finer control over who can access the router in nonprivileged and privileged mode.

With TACACS enabled, the router prompts the user for a username and a password. Then, the router queries a TACACS server to determine whether the user provided the correct password. A TACACS server typically runs on a UNIX workstation. Public domain TACACS servers can be obtained via anonymous ftp to ftp.cisco.com in the /pub directory. Use the /pub/README file to find the filename. A fully supported TACACS server is bundled with CiscoWorks Version 3.

The configuration command **tacacs-server host** specifies the UNIX host running a TACACS server that will validate requests sent by the router. You can enter the **tacacs-server host** command several times to specify multiple TACACS server hosts for a router.

Nonprivileged Access

If all servers are unavailable, you may be locked out of the router. In that event, the configuration command **tacacs-server last-resort** [**password** | **succeed**] allows you to determine whether to allow a user to log in to the router with no password (**succeed** keyword) or to force the user to supply the standard login password (**password** keyword).

The following commands specify a TACACS server and allow a login to succeed if the server is down or unreachable:

```
tacacs-server host 129.140.1.1
tacacs-server last-resort succeed
```

To force users who access the router via Telnet to authenticate themselves using TACACS, enter the following configuration commands:

```
line vty 0 4
login tacacs
```

Privileged Access

This method of password checking can also be applied to the privileged mode password with the **enable use-tacacs** command. If all servers are unavailable, you may be locked out of the router. In that event, the configuration command **enable last-resort** [**succeed** | **password**] allows you to determine whether to allow a user to log in to the router with no password (**succeed** keyword) or to force the user to supply the enable password (**password** keyword). There are significant risks to using the **succeed** keyword. If you use the **enable use-tacacs** command, you must also specify the **tacacs-server authenticate enable** command.

The **tacacs-server extended** command enables a Cisco device to run in extended TACACS mode. The UNIX system must be running the extended TACACS daemon, which can be obtained via anonymous ftp to ftp.cisco.com. The filename is xtacacsd.shar. This daemon allows communication servers and other equipment to talk to the UNIX system and update an audit trail with information on port usage, accounting data, or any other information the device can send.

The command **username <user> password** [**0** | **7**] **<password>** allows you to store and maintain a list of users and their passwords on a Cisco device instead of on a TACACS server. The number 0 stores the

password in cleartext in the configuration file. The number 7 stores the password in an encrypted format. If you do not have a TACACS server and still want to authenticate users on an individual basis, you can set up users with the following configuration commands:

```
username steve password 7 steve-pass
username allan password 7 allan-pass
```

The two users, Steve and Allan, will be authenticated via passwords that are stored in encrypted format.

Token Card Access

Using TACACS service on routers and communications servers, support for physical card key devices, or token cards, can also be added. The TACACS server code can be modified to provide support for this without requiring changes in the setup and configuration of the routers and communication servers. This modified code is not directly available from Cisco.

The token card system relies on a physical card that must be in your possession in order to provide authentication. By using the appropriate hooks in the TACACS server code, third-party companies can offer these enhanced TACACS servers to customers. One such product is the Enigma Logic SafeWord security software system. Other card-key systems, such as Security Dynamics SmartCard, can be added to TACACS as well.

Simple Network Management Protocol (SNMP) Access

SNMP is another method you can use to access your routers. With SNMP, you can gather statistics or configure the router. Gather statistics with get-request and get-next-request messages, and configure routers with set-request messages. Each of these SNMP messages has a community string that is a cleartext password sent in every packet between a management station and the router (which contains an SNMP agent). The SNMP community string is used to authenticate messages sent between the manager and agent. Only when the manager sends a message with the correct community string will the agent respond.

The SNMP agent on the router allows you to configure different community strings for nonprivileged and privileged access. You configure community strings on the router via the configuration command **snmp-server community** <string> [**RO** | **RW**] [access-list]. The following sections explore the various ways to use this command.

Unfortunately, SNMP community strings are sent on the network in cleartext ASCII. Thus, anyone who has the ability to capture a packet on the network can discover the community string. This may allow unauthorized users to query or modify routers via SNMP. For this reason, using the **no snmp-server trap-authentication** command may prevent intruders from using trap messages (sent between SNMP managers and agents) to discover community strings.

The Internet community, recognizing this problem, greatly enhanced the security of SNMP version 2 (SNMPv2) as described in RFC 1446. SNMPv2 uses an algorithm called MD5 to authenticate communications between an SNMP server and agent. MD5 verifies the integrity of the communications, authenticates the origin, and checks for timeliness. Further, SNMPv2 can use the data encryption standard (DES) for encrypting information.

Nonprivileged Mode

Use the **RO** keyword of the **snmp-server community** command to provide nonprivileged access to your routers via SNMP. The following configuration command sets the agent in the router to allow only SNMP get-request and get-next-request messages that are sent with the community string "public":

```
snmp-server community public RO 1
```

You can also specify a list of IP addresses that are allowed to send messages to the router using the access-list option with the **snmp-server community** command. In the following configuration example, only hosts 1.1.1.1 and 2.2.2.2 are allowed nonprivileged mode SNMP access to the router:

```
access-list 1 permit 1.1.1.1
access-list 1 permit 2.2.2.2
snmp-server community public RO 1
```

Privileged Mode

Use the **RW** keyword of the **snmp-server community** command to provide privileged access to your routers via SNMP. The following configuration command sets the agent in the router to allow only SNMP set-request messages sent with the community string *private*:

```
snmp-server community private RW 1
```

You can also specify a list of IP addresses that are allowed to send messages to the router by using the access-list option of the **snmp-server community** command. In the following configuration example, only hosts 5.5.5.5 and 6.6.6.6 are allowed privileged mode SNMP access to the router:

```
access-list 1 permit 5.5.5.5
access-list 1 permit 6.6.6.6
snmp-server community private RW 1
```

Controlling Access to Network Servers That Contain Configuration Files

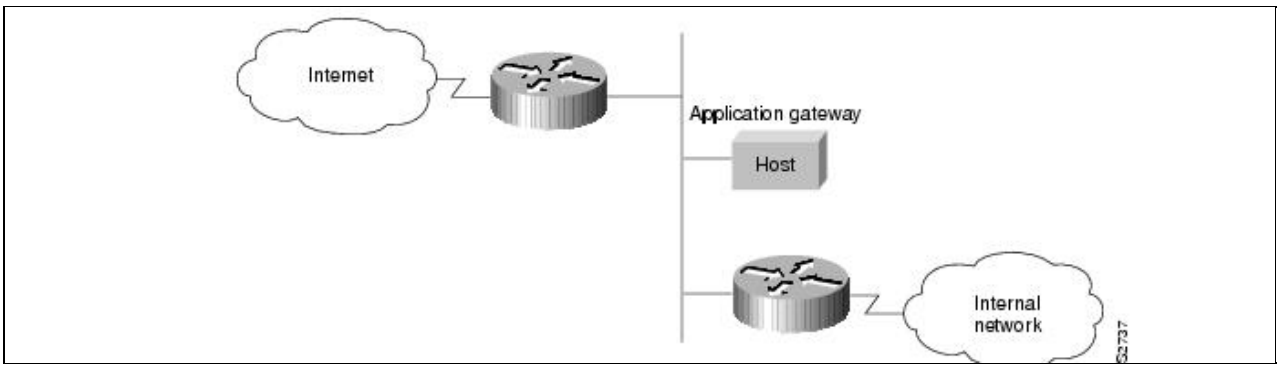
If a router regularly downloads configuration files from a Trivial File Transfer Protocol (TFTP) or Maintenance Operations Protocol (MOP) server, anyone who can access the server can modify the router configuration files stored on the server.

Communication servers can be configured to accept incoming local area transport (LAT) connections. Protocol translators and their translating router brethren can accept X.29 connections. These different types of access should be considered when creating a firewall architecture.

Setting Up Your Firewall Architecture

A firewall architecture is a structure that exists between you and the outside world to protect you from intruders. In most circumstances, intruders are represented by the global Internet and the thousands of remote networks it interconnects. Typically, a network firewall consists of several different machines as shown in Figure: Typical firewall architecture.

Figure: Typical firewall architecture



In this architecture, the router that is connected to the Internet (exterior router) forces all incoming traffic to go to the application gateway. The router that is connected to the internal network (interior router) accepts packets only from the application gateway.

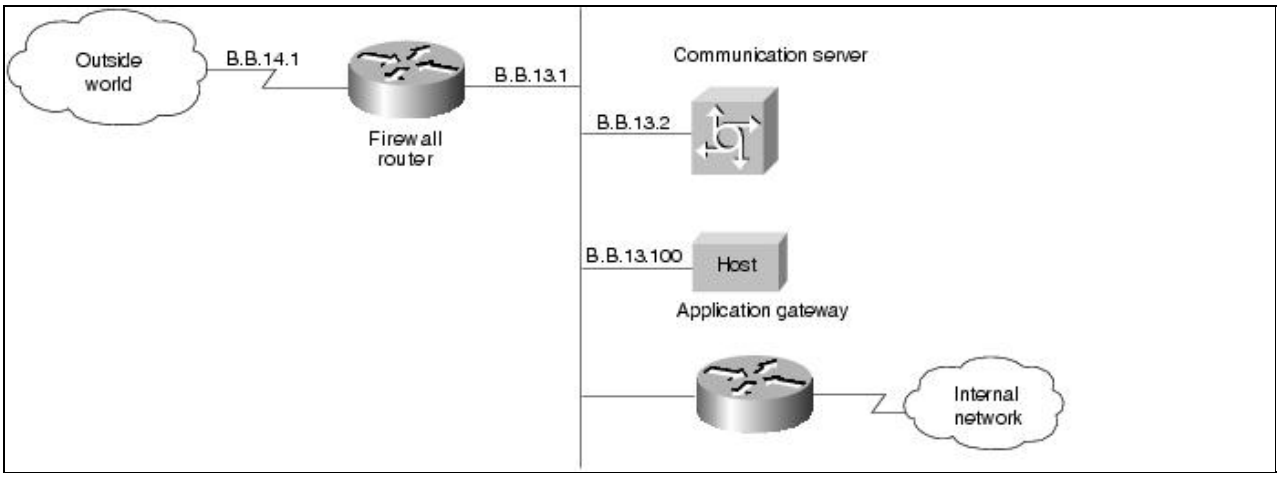
The application gateway institutes per-application and per-user policies. In effect, the gateway controls the delivery of network-based services both into and from the internal network. For example, only certain users might be allowed to communicate with the Internet, or only certain applications are permitted to establish connections between an interior and exterior host.

The route and packet filters should be set up to reflect the same policies. If the only application that is permitted is mail, only mail packets should be allowed through the router. This protects the application gateway and avoids overwhelming it with packets that it would otherwise discard.

Controlling Traffic Flow

This section uses the scenario illustrated in Figure: Controlling traffic flow via the firewall router to describe the use of access lists to restrict traffic to and from a firewall router and a firewall communication server.

Figure: Controlling traffic flow via the firewall router



In this case study, the firewall router allows incoming new connections to one or more communication servers or hosts. Having a designated router act as a firewall is desirable because it clearly identifies the router's purpose as the external gateway and avoids encumbering other routers with this task. In the event that the internal network needs to isolate itself, the firewall router provides the point of isolation so that the rest of the internal network structure is not affected.

Figure: Typical firewall architecture

Connections to the hosts are restricted to incoming file transfer protocol (FTP) requests and email services as described in the "Configuring the Firewall Router" section later in this article. The incoming Telnet, or modem, connections to the communication server are screened by the communication server running TACACS username authentication, as described in the "Configuring the Firewall Communication Server" section later in this article.

Note: Connections from one communication server modem line to another outgoing modem line (or to the outside world) should be disallowed to prevent unauthorized users from using your resources to launch an attack on the outside world. Because intruders have already passed the communication server TACACS authentication at this point, they are likely to have someone's password. It is an excellent idea to keep TACACS passwords and host passwords distinct from one another.

Configuring the Firewall Router

In the firewall router configuration that follows, subnet 13 of the Class B network is the firewall subnet, whereas subnet 14 provides the connection to the worldwide Internet via a service provider:

```
interface ethernet 0
ip address B.B.13.1 255.255.255.0
interface serial 0
ip address B.B.14.1 255.255.255.0
router igrp
network B.B.0.0
```

This simple configuration provides no security and allows all traffic from the outside world onto all parts of the network. To provide security on the firewall router, use access lists and access groups as described in the next section.

Defining Access Lists

Access lists define the actual traffic that will be permitted or denied, whereas an access group applies an access list definition to an interface. Access lists can be used to deny connections that are known to be a security risk and then permit all other connections, or to permit those connections that are considered acceptable and deny all the rest. For firewall implementation, the latter is the more secure method.

In this case study, incoming email and news are permitted for a few hosts, but FTP, Telnet, and rlogin services are permitted only to hosts on the firewall subnet. IP extended access lists (range 100 to 199) and transmission control protocol (TCP) or user datagram protocol (UDP) port numbers are used to filter traffic. When a connection is to be established for email, Telnet, FTP, and so forth, the connection will attempt to open a service on a specified port number. You can, therefore, filter out selected types of connections by denying packets that are attempting to use that service. For a list of well-known services and ports, see the "Filtering TCP and UDP Services" section later in this article.

An access list is invoked after a routing decision has been made but before the packet is sent out on an interface. The best place to define an access list is on a preferred host using your favorite text editor. You can create a file that contains the **access-list** commands, place the file (marked readable) in the default TFTP directory, and then network load the file onto the router.

The network server storing the file must be running a TFTP daemon and have TCP network access to the firewall router. Before network loading the access control definition, any previous definition of this access list is removed by using the following command:

```
no access-list 101
```

The **access-list** command can now be used to permit any packets returning to machines from already established connections. With the **established** keyword, a match occurs if the TCP datagram has the acknowledgment (ACK) or reset (RST) bits set.

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 established
```

If any firewall routers share a common network with an outside provider, you may want to allow access from those hosts to your network. In this case study, the outside provider has a serial port that uses the firewall router Class B address (B.B.14.2) as a source address as follows:

```
access-list 101 permit ip B.B.14.2 0.0.0.0 0.0.0.0 255.255.255.255
```

The following example illustrates how to deny traffic from a user attempting to spoof any of your internal addresses from the outside world (without using 9.21 input access lists):

```
access-list 101 deny ip B.B.0.0 0.0.255.255 0.0.0.0 255.255.255.255
```

The following commands allow domain name system (DNS) and network time protocol (NTP) requests and replies:

```
access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 53
access-list 101 permit udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 123
```

The following command denies the network file server (NFS) user datagram protocol (UDP) port:

```
access-list 101 deny udp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2049
```

The following commands deny OpenWindows on ports 2001 and 2002 and deny X11 on ports 6001 and 6002. This protects the first two screens on any host. If you have any machine that uses more than the first two screens, be sure to block the appropriate ports.

```
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 6001
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 6002
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2001
access-list 101 deny tcp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255 eq 2002
```

The following command permits Telnet access to the communication server (B.B.13.2):

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.2 0.0.0.0 eq 23
```

The following commands permit FTP access to the host on subnet 13:

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 21
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 20
```

For the following examples, network B.B.1.0 is on the internal network. Figure: Controlling traffic flow via the firewall router.The following commands permit TCP and UDP connections for port numbers greater than

1023 to a very limited set of hosts. Make sure no communication servers or protocol translators are in this list.

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 gt 1023
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 gt 1023
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.101 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 gt 1023
access-list 101 permit udp 0.0.0.0 255.255.255.255 B.B.1.101 0.0.0.0 gt 1023
```

Note: Standard FTP uses ports above 1023 for its data connections; therefore, for standard FTP operation, ports above 1023 must all be open. For more details, see the "File Transfer Protocol (FTP) Port" section that follows.

The following commands permit DNS access to the DNS server(s) listed by the Network Information Center (NIC):

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 53
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 eq 53
```

The following commands permit incoming simple mail transfer protocol (SMTP) email to only a few machines:

```
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.13.100 0.0.0.0 eq 25
access-list 101 permit tcp 0.0.0.0 255.255.255.255 B.B.1.100 0.0.0.0 eq 25
```

The following commands allow internal network news transfer protocol (NNTP) servers to receive NNTP connections from a list of authorized peers:

```
access-list 101 permit tcp 16.1.0.18 0.0.0.1 B.B.1.100 0.0.0.0 eq 119
access-list 101 permit tcp 128.102.18.32 0.0.0.0 B.B.1.100 0.0.0.0 eq 119
```

The following command permits Internet control message protocol (ICMP) for error message feedback:

```
access-list 101 permit icmp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Every access list has an implicit "deny everything else" statement at the end of the list to ensure that attributes that are not expressly permitted are in fact denied.

File Transfer Protocol (FTP) Port

Many sites today choose to block incoming TCP sessions originated from the outside world while allowing outgoing connections. The trouble with this is that blocking incoming connections kills traditional FTP client programs because these programs use the **PORT** command to tell the server where to connect to send the file. The client opens a "control" connection to the server, but the server then opens a "data" connection to an effectively arbitrarily chosen (> 1023) port number on the client.

Fortunately, there is an alternative to this behavior that allows the client to open the "data" socket and allows you to have the firewall and FTP too. The client sends a PASV command to the server, receives back a port

number for the data socket, opens the data socket to the indicated port, and finally sends the transfer.

In order to implement this method, the standard FTP client program must be replaced with a modified one that supports the PASV command. Most recent implementations of the FTP server already support the PASV command. The only trouble with this idea is that it breaks down when the server site has also blocked arbitrary incoming connections.

Source files for a modified FTP program that works through a firewall are now available via anonymous FTP at ftp.cisco.com. The file is /pub/passive-ftp.tar.Z. This is a version of BSD 4.3 FTP with the PASV patches. It works through a firewall router that allows only incoming established connections.

Caution: Care should be taken in providing anonymous FTP service on the host system. Anonymous FTP service allows anyone to access the hosts, without requiring an account on the host system. Many implementations of the FTP server have severe bugs in this area. Also, take care in the implementation and setup of the anonymous FTP service to prevent any obvious access violations. For most sites, anonymous FTP service is disabled.

Applying Access Lists to Interfaces

After this access list has been loaded onto the router and stored into nonvolatile random-access memory (NVRAM), assign it to the appropriate interface. In this case study, traffic coming from the outside world via serial 0 is filtered before it is placed on subnet 13 (ethernet 0). Therefore, the **access-group** command, which assigns an access list to filter incoming connections, must be assigned to Ethernet 0 as follows:

```
interface ethernet 0
ip access-group 101
```

To control outgoing access to the Internet from the network, define an access list and apply it to the outgoing packets on serial 0 of the firewall router. To do this, returning packets from hosts using Telnet or FTP must be allowed to access the firewall subnetwork B.B.13.0.

Filtering TCP and UDP Services

Some well-known TCP and UDP port numbers include the services listed in Table: Well-Known TCP and UDP Services and Ports.

Table: Well-Known TCP and UDP Services and Ports

Service	Port Type	Port Number
File Transfer Protocol (FTP)-Data	TCP	20
FTP-Commands	TCP	21
Telnet	TCP	23
Simple Mail Transfer Protocol (SMTP)-Email	TCP	25
Terminal Access Controller Access Control System (TACACS)	UDP	49
Domain Name Server (DNS)	TCP and UDP	53
Trivial File Transfer Protocol (TFTP)	UDP	69
finger	TCP	79
SUN Remote Procedure Call (RPC)	UDP	111
Network News Transfer Protocol (NNTP)	TCP	119
Network Time Protocol (NTP)	TCP and UDP	123

NeWS	TCP	144
Simple Management Network Protocol (SNMP)	UDP	161
SNMP (traps)	UDP	162
Border Gateway Protocol (BGP)	TCP	179
rlogin	TCP	513
rexec	TCP	514
talk	TCP and UDP	517
ntalk	TCP and UDP	518
Open Windows	TCP and UDP	2000
Network File System (NFS)	UDP	2049
X11	TCP and UDP	6000

CERT Advisory

The Computer Emergency Response Team (CERT) recommends filtering the services listed in Table: CERT Advisory on TCP and UDP Services and Ports.

Table: CERT Advisory on TCP and UDP Services and Ports

Service	Port Type	Port Number
DNS zone transfers	TCP	53
TFTP daemon (tftpd)	UDP	69
link-commonly used by intruders	TCP	87
SUN RPC	TCP and UDP	111
NFS	UDP	2049
BSD UNIX **r** commands (**rsh**, **rlogin**, and so forth)	TCP	512 through 514
line printer daemon (lpd)	TCP	515
UNIX-to-UNIX copy program daemon (uucpd)	TCP	540
Open Windows	TCP and UDP	2000
X Windows	TCP and UDP	6000+

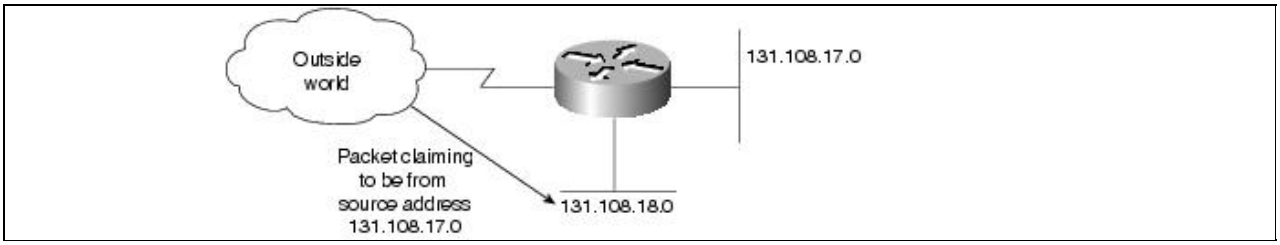
Note: Cisco recommends that you filter the finger TCP service at port 79 to prevent outsiders from learning about internal user directories and the names of hosts from which users log in.

Input Access Lists

In Software Release 9.21, Cisco introduces the ability to assign input access lists to an interface. This allows a network administrator to filter packets before they enter the router, instead of as they leave the router. In most cases, input access lists and output access lists accomplish the same functionality; however, input access lists are more intuitive to some people and can be used to prevent some types of IP address "spoofing" where output access lists will not provide sufficient security.

Figure: A host that is spoofing illustrates a host that is "spoofing," or illegally claiming to be an address that it is not. Someone in the outside world is claiming to originate traffic from network 131.108.17.0. Although the address is spoofed, the router interface to the outside world assumes that the packet is coming from 131.108.17.0. If the input access list on the router allows traffic coming from 131.108.17.0, it will accept the illegal packet. To avoid this spoofing situation, an input access list should be applied to the router interface to the outside world. This access list would not allow any packets with addresses that are from the internal networks of which the router is aware (17.0 and 18.0).

Figure: A host that is spoofing



If you have several internal networks connected to the firewall router and the router is using output filters, traffic between internal networks will see a reduction in performance created by the access list filters. If input filters are used only on the interface going from the router to the outside world, internal networks will not see any reduction in performance.

Note: If an address uses source routing, it can send and receive traffic through the firewall router. For this reason, you should always disable source routing on the firewall router with the **no ip source-route** command.

Configuring the Firewall Communication Server

In this case study, the firewall communication server has a single inbound modem on line 2:

```
interface Ethernet0
ip address B.B.13.2 255.255.255.0
!
access-list 10 deny B.B.14.0 0.0.0.255
access-list 10 permit B.B.0.0 0.0.255.255
!
access-list 11 deny B.B.13.2 0.0.0.0
access-list 11 permit B.B.0.0 0.0.255.255
!
line 2
login tacacs
location FireWallCS#2
!
access-class 10 in
access-class 11 out
!
modem answer-timeout 60
modem InOut
telnet transparent
terminal-type dialup
flowcontrol hardware
stopbits 1
rxspeed 38400
txspeed 38400
!
tacacs-server host B.B.1.100
tacacs-server host B.B.1.101
tacacs-server extended
!
line vty 0 15
login tacacs
```

Defining Access Lists

In this example, the network number is used to permit or deny access; therefore, standard IP access list numbers (range 1 through 99) are used. For incoming connections to modem lines, only packets from hosts on the internal Class B network and packets from those hosts on the firewall subnetwork are permitted:

```
access-list 10 deny B.B.14.0 0.0.0.255
access-list 10 permit B.B.0.0 0.0.255.255
```

Outgoing connections are allowed only to internal network hosts and to the communication server. This prevents a modem line in the outside world from calling out on a second modem line:

```
access-list 11 deny B.B.13.2 0.0.0.0
access-list 11 permit B.B.0.0 0.0.255.255
```

Applying Access Lists to Lines

Apply an access list to an asynchronous line with the **access-class** command. In this case study, the restrictions from access list 10 are applied to incoming connections on line 2. The restrictions from access list 11 are applied to outgoing connections on line 2.

```
access-class 10 in
access-class 11 out
```

Using Banners to Set Up Unauthorized Use Notifications

It is also wise to use the **banner exec** global configuration command to provide messages and unauthorized use notifications, which will be displayed on all new connections. For example, on the communication server, you can enter the following message:

```
banner exec ^C

 If you have problems with the dial-in lines, please send mail to helpdesk@Corporation X.com. If y
```

Securing Nonstandard Services

There are a number of nonstandard services available from the Internet that provide value-added services when connecting to the outside world. In the case of a connection to the Internet, these services can be very elaborate and complex. Examples of these services are World Wide Web (WWW), Wide Area Information Service (WAIS), gopher, and Mosaic. Most of these systems are concerned with providing a wealth of information to the user in some organized fashion and allowing structured browsing and searching.

Most of these systems have their own defined protocol. Some, such as Mosaic, use several different protocols to obtain the information in question. Use caution when designing access lists applicable to each of these services. In many cases, the access lists will become interrelated as these services become interrelated.

Summary

Although this case study illustrates how to use Cisco network layer features to increase network security on IP networks, in order to have comprehensive security, you must address all systems and layers.

Recommended Reading

This section contains a list of publications that provide internetwork security information. Books and Periodicals

Cheswick, B. and Bellovin, S. *Firewalls and Internet Security.* Addison-Wesley.

Comer, D.E and Stevens, D.L., *Internetworking with TCP/IP.* Volumes I-III. Englewood Cliffs, New Jersey: Prentice Hall; 1991-1993.

Curry, D. *UNIX System Security-A Guide for Users and System Administrators.*

Garfinkel and Spafford. *Practical UNIX Security.* O'Reilly & Associates.

Quarterman, J. and Carl-Mitchell, S. *The Internet Connection*, Reading, Massachusetts: Addison-Wesley Publishing Company; 1994.

Ranum, M. J. *Thinking about Firewalls*, Trusted Information Systems, Inc.

Stoll, C. *The Cuckoo's Egg*. Doubleday.

Treese, G. W. and Wolman, A. *X through the Firewall and Other Application Relays.*

Requests For Comments (RFCs)

RFC 1118. "The Hitchhiker's Guide to the Internet." September 1989.

RFC 1175. "A Bibliography of Internetworking Information." August 1990.

RFC1244. "Site Security Handbook." July 1991.

RFC 1340. "Assigned Numbers." July 1992.

RFC 1446. "Security Protocols for SNMPv2." April 1993.

RFC 1463. "FYI on Introducing the Internet-A Short Bibliography of Introductory Internetworking Readings for the Network Novice." May 1993.

RFC 1492. "An Access Control Protocol, Sometimes Called TACACS." July 1993.

Internet Directories

Documents at *gopher.nist.gov*. The "Computer Underground Digest" in the */pub/cud directory at ftp.eff.org.* Documents in the */dist/internet_security* directory at *research.att.com.*