WHITE PAPER

Building a Perimeter Security Solution with the Cisco Secure Integrated Software

The Value of Network Security

As different levels of Internet connectivity become essential for organizations to remain competitive, network planners place increasing emphasis on securing the network infrastructure. Organizational requirements vary, from those seeking basic Internet access to those planning to provide fundamental Web services to the outside world, creating an Internet presence. Organizations also recognize the value of developing business models that incorporate revenue generated via electronic commerce. Many companies control rising WAN line costs by basing their intranets and extranets on virtual private network (VPN) services provided via the Internet. Allowing access to sensitive resources over the inherently wide-open Internet creates an additional set of security challenges. In all these cases, network security services are key to enabling an organization to truly leverage the Internet to its competitive advantage.

Administrators can respond with a wide range of security technologies. These measures protect against not only intentional intrusions, but also against any number of common mistakes, even by authorized users.

Cisco Systems offers a complete network security solution that includes user authentication and authorization, data privacy, and perimeter security. This paper focuses on perimeter security. Perimeters most commonly exist at points where private networks meet public ones, such as a corporate Internet access point. However, perimeters can also be appropriate within an organization's network, to protect sensitive resources such as engineering workgroup servers or financial databases from unauthorized users.

Perimeter security is traditionally provided by a firewall, which inspects packets and sessions to determine if they should be transmitted or dropped. In effect, firewalls have become a single point of network access where traffic can be analyzed and controlled according to parameters such as application, address, and user, for both incoming traffic from remote users and outgoing traffic to the Internet.

Need for Cost-Effective, Integrated Firewall Solutions

Firewalls are designed to combat many kinds of attacks on network resources. Any business—including small and medium-sized—connecting to an untrusted network (such as the Internet) needs some form of firewall to protect the internal network. A security risk assessment may uncover a need for a firewall, but the risk or return may not justify the high price of a standalone firewall. Another deterrent is the complex management required to configure a firewall that enforces a company's security policy.

Cisco responded to the need for a low-cost, advanced firewall solution with the Cisco Secure Integrated Software, formerly known as the Cisco IOS® Firewall feature set, a Cisco IOS software image deployed in Cisco routers. Cisco IOS software has included necessary perimeter security capabilities in previous releases, allowing Cisco customers to use their routers as a primary point of network access control to their internal networks. The Cisco Secure Integrated Software adds practical, state-of-the-art firewall technology to Cisco IOS-based routers. As an integrated solution, it leverages what a customer already owns (a perimeter router) and understands (Cisco IOS software) to simplify ownership and management considerations. As software, it is inexpensive, readily configurable, and easily upgradable.

Types of Attack

In general, firewalls are intended to protect network resources from several kinds of attacks:

- *Passive Eavesdropping/Packet Sniffing*—Attacker uses a packet sniffer to glean sensitive information from data streams between two sites or to steal username/password combinations, either on a private carrier or a public network. Even if applications such as Lotus Notes were to encrypt traffic within their own streams, a sniffer could still detect sites using Notes in a form of traffic analysis. The attacker could then concentrate on transmissions involving that application.
- *IP Address Spoofing*—An attacker pretends to be a trusted computer by using an IP address that is within the accepted range of IP addresses for an internal network.
- *Port Scans*—An active method of determining to which ports on a network device a firewall is listening. After attackers discover the "holes" in a firewall, they can concentrate on finding an attack that exploits the applications that use those ports.
- *Denial-of-Service Attack*—Differs from other types of attack because, instead of seeking access, the attacker attempts to block valid users from accessing a resource or gateway. This blockage can be achieved through SYN flooding a network resource to exhaustion through using half-open sessions (sending TCP packets with the SYN bit set from a false address) or by crafting packets that cause a resource to perform incorrectly or crash.
- *Application-Layer Attack*—Takes many forms, exploiting weaknesses in server software to access hosts by obtaining the permission of the account that runs an application. For example, an attacker might use Simple Mail Transfer Protocol (SMTP) to compromise hosts that run older versions of sendmail using undocumented commands in the sendmail application.

Another method of attack is via a "Trojan horse," whereby the user is induced to run a malicious piece of software by being misled into believing it is something other than what it really is. More advanced application-layer attacks exploit the complexity of new technologies such as HTML, Web browser functionality, and the Hypertext Transfer Protocol (HTTP). These attacks include Java applets and ActiveX controls to pass harmful programs across a network and load them via user Web browsers.

Both the Cisco Secure Integrated Software (IS) and Cisco IOS encryption features can prevent many of these attacks. The Cisco Secure IS contains functionality to identify attacks within data streams, perform Java blocking, and limit SMTP commands that can be sent. It also contains advanced denial-of-service detection and prevention capabilities. Networks configured with Cisco IOS encryption and IPSec can protect against IP spoofing and sniffer attacks.

This paper discusses the features unique to the Cisco Secure Integrated Software, with particular attention to its key feature, the context-based access control (CBAC).

Executive Summary: Cisco Secure Integrated Software

The Cisco Secure Integrated Software introduces several enhancements to existing security services in Cisco IOS software. The most significant feature is CBAC, which enables many other new features such as Java applet blocking, Denial-of-Service detection and prevention, and audit trails.

The new feature set contains:

- *CBAC*—Provides secure, per-application access control for all IP traffic across perimeters (for example, between private enterprise networks and the Internet)
- *Java blocking*—Protects against identified malicious Java applets
- *Denial-of-Service detection/prevention*—Defends and protects router resources against SYN floods and closely related TCP attacks; checks packet headers and drops suspicious packets
- *Audit trail*—Details connections; records time stamp, source host, destination host, ports, and total number of bytes transmitted
- *Real-time alerts*—Logs alerts in case of denial-of-service attacks or other suspicious conditions
- *ConfigMaker support*—A Windows95/WindowsNT-Wizard based network configuration tool that offers step-by-step guidance through network design, addressing, and Firewall feature set implementation

Existing Cisco IOS Software Firewall Security Features

Cisco development teams built Cisco IOS software to provide significant protection against attacks and intruders, allowing Cisco customers to effectively deploy Cisco routers as perimeter security devices. Access control lists (ACLs) have been an essential part of controlling network access and have been part of the fundamental set of capabilities in Cisco routers for many years. Existing security features also enable user authentication and authorization, protect against unknown or unwelcome source/destination addresses, hide internal IP addresses from public view, track activity within and across a router, and enable administrators to implement policy-based security at perimeters.

Existing Cisco IOS firewall security features include:

- Basic and advanced traffic filtering
  - *Standard and extended ACLs*—Apply controls over access to specific network segments and define which traffic may pass through a network segment
  - *Lock and Key dynamic ACLs*—Grant temporary access through firewalls upon user identification (username/password)
- *Cisco encryption technology*—Network-layer encryption that prevents eavesdropping or tampering with data during transmission
- *IPSec encryption*—Cisco IOS software supports the recently adopted IPSEC encryption standard
- *Policy-based multiinterface support*—Provides control of user access by IP address and interface as determined by the security policy
- *Network Address Translation (NAT)*—Enhances network privacy by hiding *internal* addresses from public view; also reduces cost of Internet access by enabling conservation of registered IP addresses
- *Peer router authentication*—Ensures that routers receive reliable routing information from trusted sources
- *Event logging*—Allows administrators to track potential security breaches or other nonstandard activities on a real-time basis by logging output from system error messages to a console terminal or syslog server, setting severity levels, and recording other parameters
- *Tunneling Protocols*—Permits transmission of non-IP traffic over an IP network, which can then be secured via encryption; reduces implementation and management costs for remote branch offices and extranets; standards-based for interoperability, using encryption with any of the following protocols:
  - Generic routing encapsulation (GRE) tunneling protocol
  - Layer 2 Forwarding (L2F) protocol
  - Layer 2 Tunneling Protocol (L2TP)

A Technical Look at Packet Filtering

Packet filtering is a process that enables a router to implement a security policy based on identifiable attributes within each packet. Cisco IOS software has a rich set of access control options that can be used to filter packets. These options can be applied to inbound or outbound packets on any interface, and they add minimal latency to packets going through the host. This type of policy enforcement can stop many of the problems caused by unwanted intrusions from the public side. Packet filters can be "stateless"; that is, they do not consider TCP session state, while still providing acceptable security for many environments.

Cisco IOS software for Cisco routers generally does not maintain any information about the state of a session (for traffic traversing the box). A router will normally cache pertinent portions of a header so that subsequent packets can be fast-switched, but it will not keep track of items such as:

- How long ago was the last packet in this session transmitted?
- Are the sequence/acknowledgment numbers climbing as expected?
- Was the session initiated from the inside or outside?
- Is the session still open or has it been closed?
- What port or ports are the return data channels using?

In general, stateless packet filtering provides less scrutiny than filters that examine session states. However, because a stateless packet filter does less processing than other technologies (such as proxy servers), it is also the fastest firewall technology available, and is often implemented in perimeter hardware solutions such as Cisco IOS routers. Cisco IOS software contains features that provide several levels of standard, stateless packet filtering:

- *Standard access lists and static extended access lists*—Provide basic traffic-filtering capabilities. Configurable criteria describe which packets should be forwarded and which should be dropped, based upon each packet's network-layer information. For example, one can block all User Datagram Protocol (UDP) packets from a specific source IP address or address range. Some extended access lists can also examine transport-layer information to determine whether to forward or block packets.
- *Lock and Key (dynamic access lists)*—Provide traffic filtering with the ability to allow temporary access through the firewall for certain individuals. These individuals must first be authenticated by a username/password mechanism (usually through an attached authentication server such as a TACACS+, Remote Authentication Dial-In User Services (RADIUS), or Kerberos server). Upon user authentication, the firewall opens to allow traffic through for the associated host. Upon logout, the firewall closes the temporary opening. This scenario provides tighter, user-based control at the firewall.

CBAC, a new packet filtering mechanism introduced in the Cisco Secure Integrated Software, provides "stateful" packet filtering, which bases decisions on the state of TCP sessions. In most situations, a firewall should maintain a record of the state of each connection going through it. The Cisco Secure Integrated Software provides this auditing capability via CBAC.

Packet Filtering

Each of the fields within IP and TCP headers contains information that can be processed by the router. The portions that are usually examined for filtering are:

- IP destination address
- IP source address
- IP protocol field
- TCP source port
- TCP destination port
- TCP flags field
  - SYN alone for a request to open a connection
  - SYN/ACK for a connection confirmation
  - ACK for a session in progress
  - FIN for session termination

Various combinations of matching (or not matching) these fields can be used to support a policy. For example, if the policy is preventing SMTP sessions initiated from IP host 1.1.1.1 with a destination address of 2.2.2.2, then the packet filter would discard packets that have:

- IP destination address = 2.2.2.2
- IP source address = 1.1.1.1
- IP protocol = 6 (for TCP)
- Destination port = 25 (for SMTP)

The other fields generally do not need to be considered, although adding a check "ACK bit not set" would guard against the connection being a non-SMTP connection initiated outgoing from port 25, for example.

Similarly, packet filters exist for other protocols such as Novell IPX and Apple AppleTalk protocols, because their formats are well documented and understood.

Context-Based Access Control

CBAC is the newest form of packet filtering available as part of the Cisco Secure Integrated Software. CBAC examines not only network and transport-layer information, but also examines application-layer protocol information (such as File Transfer Protocol [FTP]) to learn about and inspect the state of TCP or UDP sessions. This mechanism dynamically creates and deletes temporary openings in the firewall by temporarily modifying access lists to change packet filtering criteria. CBAC maintains state information in its own data structures and uses that information to create the temporary entries. CBAC retains important state information that is not retained in the access list entries. The Firewall feature set inspects each packet within a data flow to ensure that the state of the session and packets themselves meet the criteria established by the user's security policy. State information is used to make intelligent permit/deny decisions. When a session closes, its temporary ACL entry is deleted, and the opening in the firewall is closed.

CBAC is initially available for Cisco 1600 and 2500 series routers. CBAC supports the two switching modes available on these platforms—fast switching and process switching. CBAC enables a Cisco router-based firewall to support protocols that involve multiple data channels created as a result of negotiations in the control channel. Many application protocols such as Telnet and SMTP use standardized "well-known" port addresses to initiate connections. Yet many Internet, Web browser, and multimedia applications that use a "well-known" port address to open an initial control connection often use different, dynamically chosen ports for data traffic. It is impossible to predict which ports these applications may use in a given connection, and some of them may use multiple channels over several ports.

Previously, network managers were compelled to block such unpredictable application traffic, rather than risk exposing internal networks to compromise. Or they were forced to protect well-known ports, but open the router to a very large range of "high number" ports to open back-paths and data channels. Using a router-based integrated firewall with CBAC, managers can protect their networks and still enable multichannel application traffic across nonwell-known ports. CBAC monitors each application on a per-connection basis for comprehensive traffic control capability. CBAC watches application sessions, notes the ports each session is using, and opens the appropriate channels only for the duration of the session, closing them when the session is finished.

How CBAC Works

- *Packets are inspected*—Arriving packets are compared to the access lists at the interfaces through which they enter and leave the router. If a packet is permitted by the existing access list, it is inspected; if the packet initiates a new connection or opens a new data channel, the access lists can be modified at this time to permit other packets related to the new connection. With CBAC, packets are inspected when they exit or enter a protected network through any interface configured for CBAC inspection.
- *State tables maintain session state information*—If a packet passing through the Cisco IOS Firewall passes CBAC inspection, the packet is forwarded, and a state table entry is created or updated to maintain session information. This scenario happens for every authorized session. A state table entry is created if the packet is either starting a new TCP connection or if the packet is the first UDP packet containing address and port information not recently seen.

Return traffic is only permitted through the firewall if the state table contains information indicating that packets belong to an existing valid session. When return traffic is detected, the state table information is updated as necessary. When a session closes or times out, the state table entry for that session is deleted.

- *UDP session traffic*—UDP is a connectionless service, so there are no actual UDP "sessions." Because CBAC operates on a session-based paradigm, it approximates sessions by examining UDP packet information and determining whether the packet is similar to other UDP packets recently seen. For generic UDP, CBAC inspects source/destination addresses and ports in IP headers or UDP packets and checks the time proximity to other UDP packets. Time proximity periods are configurable.
- *Access list entries are dynamically created and deleted to permit return traffic*—CBAC dynamically creates and deletes access lists entries at the firewall interfaces according to the information maintained in the state tables. These access list entries are inserted at the beginning of existing access lists configured on the interface allowing traffic back into the internal network. These entries are what create the temporary openings in the Cisco IOS Firewall, permitting only traffic that is part of a valid, existing session.

The fact that temporary access list entries are never saved in nonvolatile random-access memory (NVRAM) ensures that unauthorized packets do not penetrate the firewall. CBAC does not allow very many “scannable” openings to outside interfaces, and this restriction thwarts port scans. The only openings that a port scanner can find are usually those services that customers require for allowing outside sessions on the protected network, such as SMTP. Scans may still find openings to target; however, CBAC mechanisms prevent some of the best-known attacks from succeeding.

Supported Protocols

If a protocol is configured for CBAC, its traffic is inspected, state information maintained, and, in general, return packets are permitted through the firewall if they belong to a valid existing session. See a complete list of CBAC-supported protocols in Appendix A. Following is a partial list of common applications and protocols:

- FTP
- SMTP
- H.323 (such as NetMeeting or ProShare)
- Java
- Trivial File Transfer Protocol (TFTP)
- UNIX r-commands (such as remote login [r-login], remote exec [r-exec], and remote shell protocol [r-sh])
- RealAudio
- Sun RPC (not DCE RPC; not Microsoft RPC)
- The WhitePine version of CU-SeeMe
- SQL*Net
- StreamWorks
- VDOLive

Performance

Using “smart” packet inspection, the Cisco Secure Integrated Software minimizes performance impact wherever possible. With CBAC, Cisco has implemented an efficient method of monitoring traffic without significantly impacting overall network performance the way other firewall mechanisms (such as proxy servers) can. CBAC inspects and monitors only control channels; the payload of the data channels is not inspected. CBAC inspection recognizes application-specific commands in the control channel and detects and prevents application-level attacks. For example, in a NetMeeting videoconference, CBAC inspects the TCP control channel used to establish media channels. This control channel contains information that opens new media channels. CBAC watches it to identify those ports that media channels use and opens additional channels on a dynamic basis. The media and media control channels for audio and video are not inspected or monitored, because these channels only transport data and cannot open additional channels. This maximizes router and network performance to assure proper delivery of time-sensitive data.

CBAC and Encryption

CBAC and encryption are compatible on the same interface for any application-layer protocol, but it may not always be effective. Because the interactions between CBAC and encryption are complex and subject to many variables, they are worthy of separate discussion. Consequently, for the purposes of this paper they are left to other sources for further clarification.

Other New Features

Java and Other Application Blocking

With the proliferation of Java applets available on the Internet, protecting networks from malicious applets has become a major concern to network managers. HTTP connections can be configured to filter or completely deny access to Java applets that are not embedded in an archive or compressed file. Java applets within HTTP traffic are blocked based on Web server IP address. All traffic from Web servers that contain Java applets can be explicitly permitted or denied via standard access lists.

At its initial release, the Cisco Secure Integrated Software does not support ActiveX or VB script blocking.

SMTP Applications

CBAC limits SMTP applications-only standard SMTP commands. This control addresses the problem in older versions of sendmail on UNIX machines that contained undocumented commands to simplify remote troubleshooting. CBAC prevents undocumented commands from reaching a protected host.

FTP

CBAC also blocks three-way FTP sessions which can pose an unnecessary risk. CBAC improves efficiency when a user on one computer transfers data between two other computers.

Denial-of-Service Detection and Prevention

Denial-of-Service attacks differ from other types of attack because, instead of seeking access, the attacker attempts to block valid users from accessing a resource or gateway. CBAC enables enhanced, network-level D-o-S detection and prevention to defend networks against SYN flooding and packet injection.

- *SYN flooding*—Floods a network resource to exhaustion with SYN packets. This attack creates high incoming connection rates to use up available host memory or other resources or sends queries to determine whether certain ports at a firewall are listening. Attacks that involve specific network server applications such as an HTTP or FTP server operate by acquiring and keeping open all available sessions until valid users are locked out. These attacks can also degrade overall network performance by flooding the network with undesired, useless packets or by providing false status information about network resources.
- *Packet injection*—Attackers may try to send packets to a protected host that may disrupt one or many ongoing sessions. This scenario is most successful with Internet Control Message Protocol (ICMP), or by crafting packets that have sequence numbers that match an ongoing session.

Attack detection is accomplished in two ways:

- Comparing the rate of requests for new connections and the number of half-open connections to a configurable threshold level to detect SYN flooding. When the router detects unusually high rates of new connections, it issues an alert message and takes a preconfigured action as outlined following.
- All TCP connections are monitored to inspect packet sequence numbers in and detect packet injections. If the numbers are not within expected ranges, the router drops suspicious packets.

D-o-S attacks are prevented in one of two ways:

- Dropping old, half-open TCP connections to prevent system resource depletion. Telling the host to clear out old connections prevents the system from overloading or shutting down. The administrator configures a maximum threshold number for half-open connections and a timeout value before half-open connections are deleted. This method works well for low-speed connections (128 kbps or lower).
- Temporarily disabling or blocking all *SYN* packets into the host under attack to protect the router. This temporary blockage keeps the rest of the system operating, although it disables the initiating of new connections to the host. The administrator can configure an automatic timeout period for the protected server host to serve new connections again or can manually restart the router. This method works best for high-speed connections (greater than 128 kbps).

Audit Trail

CBAC information also enables enhanced audit trail features, which use syslog to track all connections: recording time stamps, source host, destination host, ports used, and the total number of transmitted bytes.

Real-Time Alerts

Real-time alerts send syslog error messages to central management consoles upon detecting suspicious activity. The feature has default settings that generate alerts upon D-o-S attacks, intrusion detection, SMTP command attacks, and denied Java applets.

ConfigMaker Support (Available Q3'98)

The Cisco Secure Integrated Software is easy to set up using ConfigMaker, a Windows95/ WindowsNT-Wizard-based management tool that enables any supported router on a network to be configured as a firewall. ConfigMaker is a configuration alternative to existing Cisco command-line interface (CLI) tools. It provides a guide through network design and router installation processes. Rather than configure each router as an independent device, ConfigMaker allows configuration of an entire network of routers—all from a single PC.

Deployment Considerations

Cisco routers with the Cisco Secure Integrated Software provide a cost-effective, integrated firewall solution. Following are several considerations for deployment within an end-to-end network:

- As an integrated, router-based firewall, fewer boxes must be managed, resulting in a less expensive solution than standalone firewalls.
- Standalone firewalls may not support the same user interfaces as other devices, so management staff must spend time learning specialized user interfaces. The Cisco Secure Integrated Software leverage a manager's existing expertise in Cisco IOS software.
- The Cisco Secure Integrated Software is initially available for Cisco 1600 and 2500 series routers. The Firewall feature will be available for additional Cisco router platforms in upcoming Cisco IOS software releases.
- As a software-based solution in low-end router platforms, the Cisco Secure Integrated Software is limited by the platform capabilities. It is most useful in small and medium-sized businesses that need to protect their Internet access points or in branch offices of large corporations using Cisco 1600 or 2500 routers to access the Internet and the main office resources.
- Integrated firewalls may also be applicable at internal perimeters within an enterprise network to protect from casual access vital assets such as financial databases or sensitive engineering projects.

Where to Use CBAC

CBAC is configurable on a per-interface basis, giving administrators discretion to apply it as needed only to interfaces that lead to untrusted networks such as the Internet. CBAC inspects standard TCP and UDP Internet applications, multimedia applications, and database applications.

In general, CBAC is deployed unidirectionally. That is, CBAC inspects outbound traffic from a private network to a public network (such as the Internet). In some cases, it may be appropriate to apply CBAC in both directions when networks on both sides of a firewall require protection (such as on an extranet or intranet). For instance, CBAC in an integrated firewall between two partner company networks can restrict traffic in one direction for certain applications and in the opposite direction for other applications.

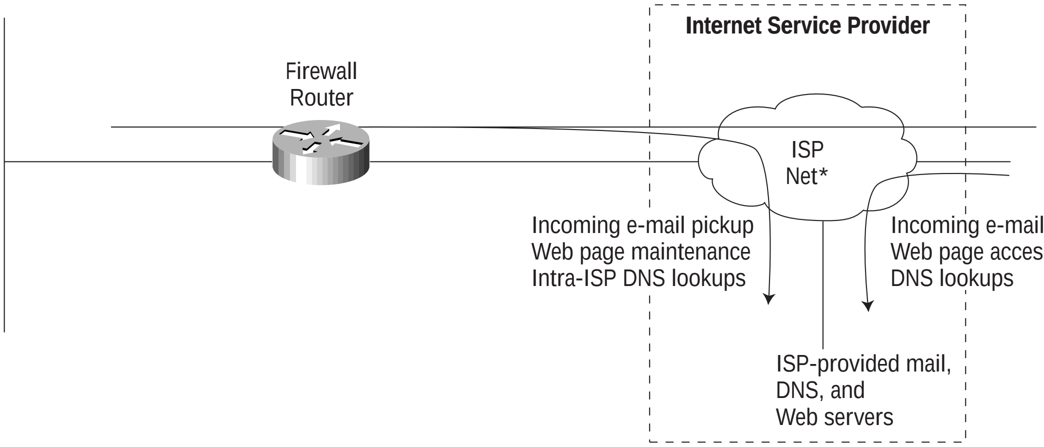
Application Scenarios

As part of an integrated Internet access router solution, the Cisco Secure Integrated Software is useful in branch offices of larger companies and main or remote offices of small and medium-sized businesses—wherever a Cisco 2500 or 1600 series router is also appropriate.

Example 1: A Home Office Network

The home office network scenario represents perhaps the simplest possible firewall configuration. It's a one-person home office, owned by an independent contractor whom we'll call Sue. Sue does most of her work using a computer; she has a LAN with several different computers connected to it. Some of those computers contain proprietary data that are important to Sue and to her clients.

Figure 1 A Home Office Network



Although Sue uses the Internet to get information she needs for her contracts, she doesn't want the trouble and expense of running her own Internet services for use by others. Like many small business Web sites, Sue's World Wide Web (www) site is on a server managed by her Internet service provider (ISP). Sue's electronic mail is delivered to a server at the ISP; Sue picks up the mail from the ISP server using the Post Office Protocol (POP)-3. Sue's domain name is a "virtual" domain hosted on the ISP's server.

Accordingly, Sue doesn't need to accept any incoming connections from the Internet to her home office LAN.

Sue configures CBAC on her Cisco 1600 router to permit only outgoing connections; the computers on her home LAN can initiate sessions with outside computers, but no outside computer can initiate a session with any of the computers on Sue's LAN. This lets her view Web pages, send e-mail, pick up incoming e-mail from her ISP, retrieve software via FTP, connect to remote systems (such as the ISP-provided server that hosts her Web pages) using Telnet, and join in multimedia conferences, all without exposing any services on her own network. Even if Sue accidentally installs a piece of software that offers a service she doesn't know about, nothing outside her own network can connect to that service.

Sue doesn't need much denial-of-service protection; she keeps the TCP/IP stacks on her computers up to date, and her link to the Internet is slow enough that a flooding attack is unlikely to be able to overrun her hosts. She is her own system administrator, so she'll be immediately aware of anything that goes wrong. Sue chooses to permit certain ICMP control traffic to enter her network for convenience in management and troubleshooting. Sue doesn't have a computer that's appropriate for use as a logging server; if she did, she could get more security information by reviewing the syslog messages generated by the router

A Cisco router configuration file implementing this scenario is given as Example 1 in the appendix to this document. Comments in that file describe some of the technical aspects of the security tradeoffs involved.

A similar configuration might be appropriate for a small branch office of a large corporation where the corporate network provides services such as e-mail servers, yet where the branch office has data that should be protected from some or all users outside of the branch.

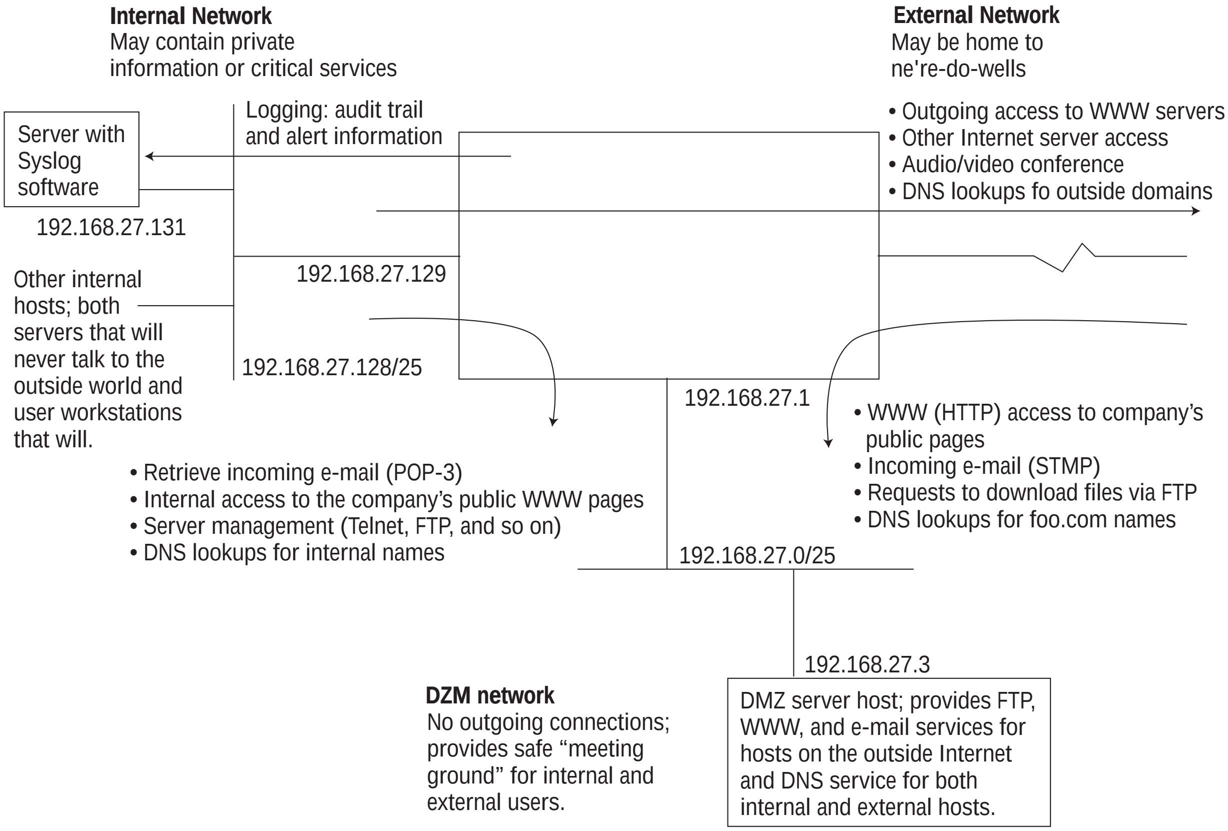
Example 2: A Small Business with a DMZ

Although our friend Sue enjoys a very simple configuration, she also gives a lot of control over her Internet services to her ISP. The second example addresses the requirements of those small and medium-sized organizations that need to offer their own Internet services to the outside world. This scenario involves a company, that we'll call FooCo. FooCo needs to administer its own www (HTTP), file transfer (FTP), electronic mail (SMTP), and name resolution (Domain name system [DNS]) services, and to make those services available over the Internet.

Although it's possible to operate publicly accessible servers within a private network, serious dangers are involved. Even with well-administered firewalls and up-to-date host software, there's always a chance that attackers could use an application-level attack to gain unrestricted access to a computer that is running a server program. If that happens when the server computer is inside a private network, the attacker may gain access to important private data or greatly disrupt the operation of critical internal services.

Because of the dangers of running servers inside private networks, many firewall configurations, including FooCo's, employ "DMZ" (demilitarized zone) networks. A DMZ network is intended to provide a safe, relatively neutral "drop area" for communication between inside and outside systems. The network topology is shown in Figure 2.

Figure 2 A Small Business with a DMZ



Internal Network
May contain private information or critical services
External Network
May be home to ne're-do-wells
Server with Syslog software
192.168.27.131
Logging: audit trail and alert information
• Outgoing access to WWW servers
• Other Internet server access
• Audio/video conference
• DNS lookups fo outside domains
192.168.27.129
Other internal hosts; both servers that will never talk to the outside world and user workstations that will.
192.168.27.128/25
192.168.27.1
• WWW (HTTP) access to company's public pages
• Incoming e-mail (STMP)
• Requests to download files via FTP
• DNS lookups for foo.com names
• Retrieve incoming e-mail (POP-3)
• Internal access to the company's public WWW pages
• Server management (Telnet, FTP, and so on)
• DNS lookups for internal names
192.168.27.0/25
192.168.27.3
DZM network
No outgoing connections; provides safe "meeting ground" for internal and external users.
DMZ server host; provides FTP, WWW, and e-mail services for hosts on the outside Internet and DNS service for both internal and external hosts.

No connections are permitted from the untrusted Internet to the private network; instead, Internet users connect to servers on the DMZ network. This server provides all the services that FooCo wishes to offer to outside users. The DMZ server itself is prevented from initiating connections into the private network; although systems on the private network can use the services provided by the DMZ, the DMZ cannot reach internal services.

Because connections cannot be made from the DMZ server to hosts on the internal network even if attackers can break into the DMZ server, the damage they can cause is limited. They may be able to embarrass FooCo, for example, by modifying its WWW pages, but they are unable to access company internal data after breaking into the DMZ server.

As a further refinement, FooCo takes advantage of the fact that no reason exists for the DMZ server ever to initiate an outgoing connection to the Internet. If attackers penetrate the DMZ server, they might use it as a tool to attack other Internet sites, which could create a variety of problems for FooCo and others. To prevent this scenario, FooCo administrators configure the firewall to prevent connections from the DMZ network, not only to services on the private network, but also to services on the outside Internet.

An unusual choice in this example is to place the sole DNS server in the DMZ. Because DNS is a critical service for IP networks, it is more common to have two separate DNS servers (or two separate *sets* of redundant servers): one on the private network for use by inside hosts, and another on the DMZ network for use by outside hosts. This approach has better security than the single-server approach if it is properly implemented, but FooCo has decided to forego that extra measure of security for the administrative convenience of having a single primary server.

FooCo allows its internal users to make unrestricted connections to the Internet; business needs demand that users be able to access Internet resources conveniently, and the administrative demands of getting every possible service would be prohibitive. This scenario is a common security posture, and usually a justified one, but it is not without risks. An attacker on the Internet could potentially set up a server that exploits application-layer vulnerabilities in user/client software and use those vulnerabilities to gain unauthorized access to the private network. WWW browsers, as large, complex programs, are especially vulnerable to this sort of attack. In addition, untrained users could accidentally download and install viruses or other malicious programs. FooCo chooses to accept these risks for business reasons. It moderates the risks by keeping host software, especially Web browsers, up to date and by educating its employees about security risks.

The most security-relevant technical aspects of the FooCo configuration file are given in Example 2 in Appendix B.

Summary

The Cisco Secure Integrated Software provides an intelligent, integrated firewall solution on Cisco 1600 or 2500 series routers, ideal for small and medium-sized businesses or enterprise branch offices. As part of Cisco IOS security services, the Firewall feature set in Cisco routers cost-effectively protects network perimeters that seamlessly interoperate with other Cisco IOS software features, with minimal performance impact. Residing in Cisco routers and switches, Cisco IOS software and technologies make possible an end-to-end Cisco network that enables advanced networking capabilities and supports the latest networked applications. Additional literature about Cisco IOS security services and Cisco firewall solutions are available at http://www.cisco.com/go/security.

Appendix A: Application Protocols Supported by CBAC[JB3]

Protocol	Description
Audio/Video Streaming	
CU-SeeMe by White Pine	Application that supports live audio/videoconferencing and text chat across the Internet
H.323	New standard in audio/videoconferencing
Internet Phone by Intel	Voice communication application above H.323 protocol stack
NetMeeting by Microsoft	Audio, video, and application sharing implemented over T.120 and H.323
RealAudio and RealVideo by Progressive Networks	Protocol for the transmission of high-quality streaming sound and video on the Internet
StreamWorks by Xing	Protocol for the transmission of high-quality streaming sound and video on the Internet
VDOLive by VDOnet	Application for transmitting high-quality video over the Internet
Information Seeking	
Archie	Standard tool for searching Internet file servers
Gopher	Application that provides a menu-driven front-end to Internet services
HTTP	Primary protocol used to implement the WWW
Network News Transfer Protocol (NNTP)	Protocol used to transmit and receive network news
Pointcast by Pointcast (HTTP)	Protocol for viewing news in TV-like fashion
Wide Area Information Servers (WAIS)	Tool for keyword searches, based on database content, of databases on the Internet
Security and Authentication	
HTTPS	Secured (that is, encrypted) HTTP; an implementation of SSL
TACACS+	Authentication protocol
Kerberos	Authentication service
LDAP	Standard for Internet directory services
RADIUS	A widely adopted authentication protocol
Secure ID	Protocol used by an authentication service product of Security Dynamics Technologies, Inc.
Databases	
Lotus Notes	Proprietary protocol developed by Lotus to implement its Notes application
SQL Server by Microsoft	A data replication server
SQLNet version 1	Oracle protocol for transmission of SQL queries
SQLNet version 2	Extension of SQLNet Version 1, which adds support for port redirection
Mail	
Comsat	Mail notification protocol
Imap	Internet mail access protocol
POP Version 2	Mail protocol that allows a remote mail client to read mail from a server
POP Version 3	Modified version of POP Version 2
SMTP	Protocol widely used for the transmission of e-mail
Other TCP and UDP Services	
Chargen	TCP chargen server sends a continual stream of characters until the client terminates the connection; UPD chargen servers send a datagram containing a random number of characters in response to each datagram sent by a client
Daytime	Daytime server returns the date and the time of day in text format and can be run over TCP or UDP
Discard	Discard server discards whatever is sent to it by a client, and can be run over TCP or UDP
DNS	Distributed database used by TCP/IP to map names to IP addresses
Finger	Protocol that provides information about users on a specified host
FTP	Protocol for copying files between hosts

Protocol	Description
Identd (auth)	Protocol used for user identification
Internet Relay Chat (IRC)	Protocol for online “chat” conversations over the Internet
NetBIOS over TCP/IP (NBT)	NetBIOS name, datagram, and session services encapsulated within TCP/IP
Network Time Protocol (NTP)	Protocol providing time synchronization across a network with precise clocks, implemented over TCP and UDP
RAS	Remote access service
Rexec	Protocol that provides remote execution facilities
Rlogin	Protocol that enables remote login between hosts
Rsh	Protocol that allows commands to be executed on another system
Simple Network Management Protocol (SNMP)	Protocol used for managing network resources
SNMP Trap	Notification by an SNMP to the manager of some event of interest
Syslog	Protocol that allows a computer to send logs to another computer
Telnet—Telecommunications Network Protocol	Remote terminal protocol enabling any terminal to log in to any host
TFTP	Small, simple FTP used primarily in booting diskless systems
Time	Service that returns the time of day as a binary number
UNIX-to-UNIX Copy Program (UUCP)	UNIX file-copying protocol
Who	Service that uses local broadcasts to provide information about who is logged on to the local network
X11	Windowing system protocol
Remote Procedure Call Services	
Lockmanager (nlockmgr)	Protocol used for the transmission of lock requests
Mountd	Protocol used for the transmission of file mount requests
Network File System (NFS)	Protocol that provides transparent file access over a network
Network Information Service (NIS)	Protocol that provides a network-accessible system-administration database, widely known as Yellow Pages
Rstat	Protocol used to obtain performance data from a remote kernel
Rwall	Protocol used to write to all users in a network

Appendix B: Configuration File Examples

This appendix contains examples of the configuration commands needed to implement the scenarios in Section VII.

Example 1: A Home Office

This is the configuration file for the home office scenario described in Section VII. This is a complete configuration, based on one that's actually in use in a real site. It includes not only the commands that control security, but all the other commands necessary to configure the router. The file has been lightly edited to remove sensitive information and nonessential commands.

General notes:

- This configuration uses only input access lists, allowing it to perform both anti-spoofing and traffic filtering from the same access list.
- Ethernet 0 is the "inside" net. Serial 0 is a Frame Relay link to the ISP.

These first three commands should be in virtually every router. The small servers commands can be used to assist a variety of attacks if they are not disabled.

```
service password-encryption
no service udp-small-servers
no service tcp-small-servers
```

Define the router name.

```
hostname sue-router
```

Define the source from which to load Cisco IOS software.

```
boot system flash c1600-fw1600-l
```

Set a password to control access to the privileged command mode. Note the use of enable secret instead of enable password; enable secret is cryptographically much more secure and should be used in all new configurations.

```
enable secret 5 $1$234lkoasufnilwrnasi0o9u23nt
```

Establish a username and password, so that the administrator can log into the router.

```
username sue password 7 14191D1815023F2036
```

Permit the use of zero subnets. See the Cisco IOS software documentation for more information.

```
ip subnet-zero
```

IP source routing is almost never used for legitimate purposes and can sometimes be used to transport packets to parts of the network from which they should be blocked. Although it's not a concern for this particular scenario, configuring no ip source-route should be a habit for anyone who sets up security-sensitive routers.

```
no ip source-route
```

Define the default domain name and the servers to be used for DNS lookups initiated by the router.

```
ip domain-name sue.com
ip name-server 172.19.2.132
ip name-server 172.19.30.32
```

Configure the CBAC parameters (the configuration language refers to CBAC as inspection). These commands enable inspection for specific protocols and define session timeouts.

```
ip inspect name myfw cuseeme timeout 3600
ip inspect name myfw ftp timeout 3600
ip inspect name myfw http timeout 3600
ip inspect name myfw rcmd timeout 3600
ip inspect name myfw realaudio timeout 3600
ip inspect name myfw smtp timeout 3600
ip inspect name myfw tftp timeout 30
ip inspect name myfw udp timeout 15
ip inspect name myfw tcp timeout 3600
```

Define parameters for the internal LAN interface.

```
interface Ethernet0
description Ethernet LAN chez Sue
```

Configure IP addressing for the LAN.

```
ip address 172.19.139.1 255.255.255.248
ip broadcast-address 172.19.139.7
```

Define which access list controls traffic incoming on this interface.

```
ip access-group 101 in
```

Directed broadcasts can be used to multiply the power of denial-of-service attacks and should be disabled on any interface where they're not actually needed.

```
no ip directed-broadcast
```

Inspection is applied to packets *incoming* on the private interface, so temporary access list entries are created in response to conversations initiated *from* the private network. The temporary entries will be created in the access lists for both interfaces, even though inspection is configured only on one interface; all that's necessary is that inspection be applied to the packet that initiates each session. No outgoing inspection on this interface is needed, because the very limited ICMP data permitted to come in spontaneously from the Internet will pass access lists without inspection. If certain incoming services were allowed (for example, if Sue wanted to receive incoming Cu-SeeMe calls), then we would need to apply either outgoing inspection on this interface or incoming inspection on the Frame Relay interface.

```
ip inspect myfw in
```

It's a good idea to turn off Cisco Discovery Protocol (CDP) on any security device, on the principle that a disabled service is usually safer than an enabled one. CDP gives away some information about the router.

```
no cdp enable
```

Define parameters for the Frame Relay interface:

```
interface Serial0
```

General configuration; see the Cisco IOS software documentation for details on these commands:

```
description FR (Telco ID 79YGTQ12739-924) to ISP
no ip address
encapsulation frame-relay IETF
no arp frame-relay
bandwidth 56
service-module 56 clock source line
service-module 56k network-type dds
frame-relay lmi-type ansi
```

Define the Frame Relay subinterface over which IP traffic to and from the ISP passes:

```
interface Serial0.1 point-to-point
```

Addressing; see Cisco documentation:

```
    ip unnumbered Ethernet0
```

Choose which access list is to control traffic coming into the router (and into the private network) from the ISP. Note that, although there is no inspection on serial 0.1, the inspection on the Ethernet interface adds temporary entries to this list when hosts on the private LAN make outgoing connections through the WAN link.

```
    ip access-group 111 in
```

Generic configuration and commands already discussed:

```
    no ip directed-broadcast
    bandwidth 56
    no cdp enable
    frame-relay interface-dlci 16
```

Generic IP routing configuration:

```
    ip classless
    ip route 0.0.0.0 0.0.0.0 Serial0.1
```

Access list 101 controls what packets can be received from the private Ethernet interface. The goal of this configuration is to control what can be sent to the private interface from the Internet. Therefore, we generally assume that anything sent from a host on the private LAN is acceptable.

However, it's important to at least check that the IP packets coming from the private LAN have source addresses consistent with that LAN. This precaution avoids problems that could be caused by accidental misconfiguration, as well as some problems that might happen if somebody actually did manage to break into the private network.

```
    access-list 101 permit tcp 172.19.139.0 0.0.0.7 any
    access-list 101 permit udp 172.19.139.0 0.0.0.7 any
    access-list 101 permit icmp 172.19.139.0 0.0.0.7 any
    access-list 101 deny ip any any
```

Access list 111 controls what can come from the outside world over the Frame Relay interface.

It's important to do antispoofing in any access list that faces the Internet. Antispoofing entails checking that incoming packets don't have source addresses that claim to be from our own network, or from reserved loopback addresses, or from multicast addresses. Such packets are always the result of a misconfiguration or of a deliberate attack.

```
    access-list 111 deny ip 127.0.0.0 0.255.255.255 any
    access-list 111 deny ip 224.0.0.0 31.255.255.255 any
    access-list 111 deny ip 172.19.139.0 0.0.0.7 any
```

Permit some ICMP packets. Note that there is some risk in these; they are control packets, and allowing any packet opens the firewall to some attacks such as Teardrop-style fragmentation attacks. However, it can be difficult to administer a network properly without allowing some ICMP packets. If Sue had a higher bandwidth Frame Relay link, she might be more concerned about being flooded with some of these ICMP packets.

"Administratively prohibited" messages may explain why failing pings, traceroutes, and other things are not getting through.

```
    access-list 111 permit icmp any 172.19.139.0 0.0.0.7 administratively-prohibited
```

Outside administrators may need to ping hosts on the internal network as part of debugging. Enable echo request packets.

```
    access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo
```

Echo reply packets are needed for us to ping outside hosts. Again, this is an important debugging aid.

```
    access-list 111 permit icmp any 172.19.139.0 0.0.0.7 echo-reply
```

“Packet too big messages are needed to support maximum transmission unit (MTU) discovery. Without them, connections to some sites might run slowly or even hang.

```
    access-list 111 permit icmp any 172.19.139.0 0.0.0.7 packet-too-big
```

Time-exceeded messages allow traceroute to work from inside the network.

```
access-list 111 permit icmp any 172.19.139.0 0.0.0.7 time-exceeded
```

“Unreachables” have a few risky implications, but sometimes may help outgoing connections fail quickly, rather than forcing the user to wait for them to time out.

```
    access-list 111 permit icmp any 172.19.139.0 0.0.0.7 unreachable
```

This deny is the default, but let’s make it clear.

```
    access-list 111 deny   ip any any
```

By turning off CDP both for the entire router and for each interface, we make it less likely that it will be turned on again by accident at a later time.

```
    no cdp run
```

Parameters for the Console Port and the Virtual TTYs

We need to configure these so that it’s possible to log in to the router and so that it’s impossible to log in from the console port without a password (or physical access to the router itself).

```
    line con 0
```

It would probably be safer to have a nonzero EXEC timeout.

```
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    line vty 0
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    line vty 1
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    line vty 2
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    line vty 3
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    line vty 4
    exec-timeout 0 0
    password 7 14191D1815023F2036
    login local
    scheduler interval 500
    end
```

Example 2: A Small Business with a DMZ

This example shows a typical configuration for a small business connected with its own servers to the Internet. It corresponds to the second scenario in Section VII of this document.

Unlike the previous example, this one doesn't include the entire router configuration file; instead, only the commands relevant to firewall setup are shown. The commands are shown in logical order, rather than in the order in which they would be presented by a router. Furthermore, this configuration is untested; although it has been carefully reviewed, it is simply an example and has never actually been installed on a router.

The security policy is formally described as follows:

1. Allow inside users to connect to any service on the public Internet. The implicit assumption is that client programs are secure enough to make this reasonably safe.
2. Allow anyone on the Internet to connect to WWW, FTP, and SMTP services on the DMZ server and to make DNS queries to it. This scenario allows outside users to view FooCo's Web pages, pick up files that the company has posted for outside availability, and send mail into the company.
3. Allow inside users to connect to the POP service on the DMZ server (to pick up their mail) and to Telnet to it for administration, as well as to use the DMZ services available to the rest of the Internet.
4. Allow nothing on the DMZ to initiate any connections, either to the private network or to the Internet.
5. Allow ancillary control traffic, such as ICMP, on an as-needed basis.
6. Audit all connections crossing the firewall to a Syslog server on the private net. Full auditing of all connections consumes significant, although not usually prohibitive, network resources. It is therefore recommended only when the data is actually of potential use.

For the purposes of the configuration commands in this example,

- The Internet is connected to Serial 0.
- Ethernet 0 is connected to the internal network, which consists of a single LAN.
- Ethernet 1 is connected to the DMZ network.
- A single node on the DMZ network, at 192.169.27.3, provides all services to the outside world.
- The ISP has assigned the company the netblock 192.168.27.0/24, which has been split equally between the DMZ and the internal LAN using a subnet mask of 255.255.255.128.
- Input access lists are used on all interfaces to prevent spoofing.
- Output access lists are used to control what traffic may be sent to any given interface.
- The access list numbering convention uses the second digit of the access list number to identify the interface on which the list is used. The third digit is set to "1" for input access lists and "2" for output access lists.
- Incoming CBAC inspection is always used in preference to outgoing inspection.

Security Boilerplate

These commands, or their equivalents, should appear in almost every configuration. See the detailed discussion of these commands in Example 1 for more information.

```
no service tcp-small-servers
no service udp-small-servers
service password-encryption
no cdp running
no ip source-route
enable secret <something>
interface ethernet 0
ip address 192.168.27.129 255.255.255.128
no ip directed-broadcast
```

```
interface ethernet 1
ip address 192.168.27.1 255.255.255.128
no ip directed-broadcast
interface serial 0
```

Require login on all lines; this is one possible way.

```
line 0 6
password <something>
login
```

Inspection Parameters

Access is controlled by access lists, not by what's listed in this inspection set. The purpose of configuring an inspection set is to set parameters such as timeouts and to define the protocols to which CBAC should pay attention.

The next two commands define a logging server and request that all connections inspected by CBAC be logged.

```
logging 192.168.27.131
ip inspect audit-trail
```

The following commands configure default timeout periods. They're given for illustrative purposes only; in most cases, the system defaults are acceptable.

Time out idle TCP connections after 14,400 seconds

```
ip inspect tcp idle-time 14400
```

Time out idle UDP connections after 1,800 seconds. It's generally unwise to have an extremely long UDP timeout.

```
ip inspect udp idle-time 1800
```

DNS connections should time out much more quickly than other UDP connections. Otherwise, CBAC will have to maintain many useless connection state table entries for long-ago-completed DNS requests.

```
ip inspect dns-timeout 15
```

The next few commands turn on inspection for every protocol (except RPC, which must be enabled for each individual program number). Note that this doesn't control what connections can be created, so much as it controls how created sessions are handled. Access lists applied to the interfaces control what sessions can actually be created.

```
ip inspect name standard cuseeme
ip inspect name standard ftp
ip inspect name standard h323
ip inspect name standard http
ip inspect name standard rcmd
ip inspect name standard realaudio
ip inspect name standard smtp
ip inspect name standard sqlnet
ip inspect name standard streamworks
ip inspect name standard tcp
ip inspect name standard tftp
ip inspect name standard udp
ip inspect name standard vdolive
```

The Private Interface

Configuring the private network interface is quite simple, because there are no incoming services. This configuration has one feature that may annoy users: some servers on the Internet will try to make IDENTD connections back to clients before they process any commands. This configuration does not allow the IDENTD requests back into the private network, and some (probably most) TCP stacks will not notice the "administratively prohibited" packets that are sent back when the access list denies the SYN packet. Users may see very long delays while IDENTD connections time out when they visit certain Web sites. We could avoid this by allowing IDENTD through, but only at the cost of some security.

```
interface ethernet 0
ip address 192.168.27.129 255.255.255.128
```

Access lists for packet filtering. These lists will be dynamically modified by CBAC.

```
ip access-group 101 in
ip access-group 102 out
```

Apply CBAC to connections initiated by packets arriving on this interface (that is, to connections initiated from within the private network). This application is done with the following command:

```
ip inspect standard in
```

Access list 101 (like all input lists in this configuration) is purely antispoofing. Its purpose is to keep hosts on the private network from sending packets with source addresses on other networks. The first command allows packets with correct addresses; the second blocks all other packets.

```
access-list 101 permit ip 192.168.27.128 0.0.0.127 any
access-list 101 deny ip any any
```

Access list 102 controls incoming connections to the private net from the Internet and the DMZ. This allows no actual connections, but does allow some miscellaneous ICMP packets to make the network easier to administer and to avoid waiting for timeouts in some circumstances. Inspection will add entries to this list that permit return traffic for any outgoing connection established from the private network. See Example 1 for the reasons for allowing each of the ICMP packet types.

```
access-list 102 permit icmp any any administratively-prohibited
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
access-list 102 permit icmp any any packet-too-big
access-list 102 permit icmp any any time-exceeded
access-list 102 permit icmp any any unreachable
access-list 102 deny ip any any
```

The DMZ Interface

No "ip inspect" command exists on this interface, because outgoing conversations are never initiated from the DMZ. Note that this really means that no outgoing connections are permitted. A very common configuration mistake is to configure the DMZ server to perform reverse DNS or IDENTD lookups on client addresses. If a server uses reverse DNS or IDENTD, the firewall must be configured to permit those protocols; otherwise, users must wait for the requests to time out before they can access data. Neither reverse DNS nor IDENTD is generally very useful from a security standpoint, and each consumes more network resources than it is usually worth. Neither is recommended for most configurations. Note that the timeout problem here is exactly the same problem as described previously under "The Private Interface," but seen from the other side of the connection.

Basic configuration commands:

```
interface ethernet 1
ip address 192.168.27.1 255.255.255.128
ip access-group 111 in
ip access-group 112 out
```

Access list 111 is purely an antispoofing list. Allow only packets with source addresses on the subnet that are actually connected to this interface.

```
access-list 111 permit ip 192.168.27.0 0.0.0.127 any
access-list 111 deny ip any any
```

Access list 112 controls what traffic can enter the DMZ network and, therefore, what connections can be established to servers on that network. In this example, only one server host is on the DMZ network and 192.168.27.3.

Permit DNS lookups from any host.

```
access-list 112 permit udp any host 192.168.27.3 eq domain
```

Also permit DNS zone transfers. This would most likely be used by a backup DNS server, perhaps at an ISP site.

```
access-list 112 permit tcp any host 192.168.27.3 eq domain
```

A few services are available to the whole world.

```
access-list 112 permit tcp any host 192.168.27.3 eq www
access-list 112 permit tcp any host 192.168.27.3 eq ftp
access-list 112 permit tcp any host 192.168.27.3 eq smtp
```

Telnet and POP services are available only to hosts on the private network. Note that the only reason it is safe to filter based on source addresses in these two commands is that proper antispoofing exists in input lists for all interfaces; otherwise, anybody could spoof these addresses.

```
access-list 112 permit tcp 192.168.27.128 0.0.0.127 host 192.168.27.3 eq pop3
access-list 112 permit tcp 192.168.27.128 0.0.0.127 any eq telnet
```

ICMP boilerplate:

```
access-list 112 permit icmp any any administratively-prohibited
access-list 112 permit icmp any any echo
access-list 112 permit icmp any any echo-reply
access-list 112 permit icmp any any packet-too-big
access-list 112 permit icmp any any time-exceeded
access-list 112 permit icmp any any unreachable
```

Deny everything else. This is the default behavior for any nonempty access list, but making it explicit can improve configuration readability.

```
access-list 112 deny ip any any
```

The Internet Interface

Incoming inspection occurs on this interface. This inspection is necessary so that CBAC can place “holes” in the output access list to permit return traffic from the DMZ for connections established to the DMZ from the Internet.

```
interface serial 0
ip unnumbered ethernet 1
ip inspect standard in
ip access-group 121 in
ip access-group 122 out
```

Antispoofing is especially critical on any interface that faces the Internet. Anything not explicitly assigned to the internal net is assumed to be on the Internet, but under no circumstances should any system accept internal source addresses on packets originating outside the internal network.

Deny internal addresses:

```
access-list 121 deny ip 192.168.27.0 0.0.0.255 any
```

Deny loopbacks (such packets should never appear on any physical cable):

```
access-list 121 deny ip 127.0.0.0 0.255.255.255 any
```

Deny multicast sources:

```
access-list 121 deny ip 224.0.0.0 31.255.255.255 any
access-list 121 permit ip any any
```

The only control placed on Internet connections is to disallow outgoing connections from the DMZ. This disallowing prevents somebody who penetrates the DMZ server from using it as a launchpad for attacking other parts of the network. Note that CBAC makes holes in this list to permit return traffic from the DMZ for connections established from outside. It is usually worth the risk to permit pings and trace routes into the DMZ, so allow their replies to go back out to the Internet.

```
access-list 122 permit icmp any any echo-reply
access-list 122 permit icmp any any time-exceeded
access-list 122 deny ip 192.168.27.0 0.0.0.127 any
access-list 122 permit ip any any
```

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems Europe s.a.r.l.
Parc Evolic, Batiment L1/L2
16 Avenue du Quebec
Villebon, BP 706
91961 Courtaboeuf Cedex
France
http://www-europe.cisco.com
Tel: 33 1 6918 61 00
Fax: 33 1 6928 83 26

Americas
Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Headquarters
Nihon Cisco Systems K.K.
Fuji Building, 9th Floor
3-2-3 Marunouchi
Chiyoda-ku, Tokyo 100
Japan
http://www.cisco.com
Tel: 81 3 5219 6250
Fax: 81 3 5219 6001

Cisco Systems has more than 200 offices in the following countries. Addresses, phone numbers, and fax numbers are listed on the

Cisco Connection Online Web site at http://www.cisco.com.

Argentina • Australia • Austria • Belgium • Brazil • Canada • Chile • China (PRC) • Colombia • Costa Rica • Czech Republic • Denmark
England • France • Germany • Greece • Hungary • India • Indonesia • Ireland • Israel • Italy • Japan • Korea • Luxembourg • Malaysia
Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines • Poland • Portugal • Russia • Saudi Arabia • Scotland •
Singapore